

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-327815

(43)Date of publication of application : 30.11.1999

(51)Int.Cl. G06F 3/12
 G06F 13/00
 G06F 13/00
 H04L 29/08

(21)Application number : 10-133994

(71)Applicant : CANON INC

(22)Date of filing : 15.05.1998

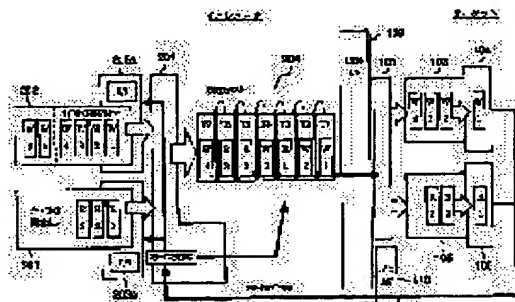
(72)Inventor : ISODA TAKASHI
 SHIMURA AKIHIRO

(54) COMMUNICATION CONTROL METHOD/DEVICE AND COMMUNICATION SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a whole duplex communication line by which the resource of a printer is required to be less and restoration after a communication interruption is guaranteed.

SOLUTION: An initiator holds a command corresponding to ORB issued to a target in I/O request queues 202 and 203 until a completion response is given from the target. The target has a write execution agent 104 and a read execution agent 106 and processes the command from the initiator. When a connection between both agents is cut and it is connected again, the initiator deletes all ORB, generates ORB again based on the held command of the I/O request queue and issues it to the target. The target holds the identifier of ORB during a processing and the address of a buffer reading or writing data. The held identifier is compared with the identifier of ORB which is issued again from the initiator after reconnection. When they are matched, reading or writing is resumed from the held address. When they are not matched, ORB is considered to be processed, and a processing completion status is returned to the initiator.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision
of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japan Patent Office

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-327815

(43)公開日 平成11年(1999)11月30日

(51)Int.Cl.⁸
 G 0 6 F 3/12
 13/00
 H 0 4 L 29/08

識別記号
 3 5 1
 3 5 7

F I
 G 0 6 F 3/12
 13/00
 H 0 4 L 13/00

A
 3 5 1 A
 3 5 7 A
 3 0 7 Z

審査請求 未請求 請求項の数19 O L (全 32 頁)

(21)出願番号 特願平10-133994

(22)出願日 平成10年(1998)5月15日

(71)出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 磯田 隆司

東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内

(72)発明者 志村 明弘

東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内

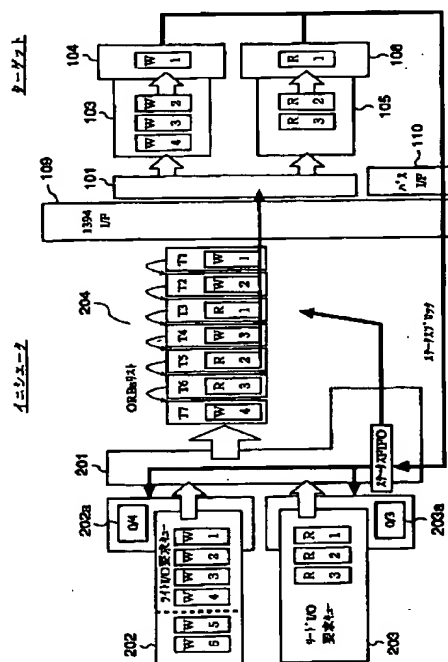
(74)代理人 弁理士 大塚 康徳 (外2名)

(54)【発明の名称】 通信制御方法及び装置及び通信システム

(57)【要約】

【課題】プリンタの資源が少なくても済み、通信切断後の復旧が保証されたむ全2重通信路を提供する。

【解決手段】イニシエータは、ターゲットに対して発行したORBに対応するコマンドを、I/O要求キュー202、203の中に、ターゲットから完了応答があるまで保持する。ターゲットは、ライト実行エージェント104とリード実行エージェント106とを有し、イニシエータからのコマンドを処理する。両者の間の接続が断たれ、再び接続されると、イニシエータは全てのORBを消去した後、I/O要求キューの保持されたコマンドをもとに、再度ORBを生成してターゲットに発行する。ターゲットは、ORBの処理時には、処理中のORBの識別子と、読み出しあるいは書き込みを行っているバッファのアドレスを保持している。再接続時には、保持している識別子と、イニシエータから再発行されたORBの識別子とを比較し、一致していれば、保持されているアドレスから読み出しあるいは書き込みを再開する。一致していなければ、そのORBは処理済みとみなして、処理完了ステータスをイニシエータに返す。



【特許請求の範囲】

【請求項1】 イニシエータからターゲットに対して命令を発行することで、前記イニシエータの有する記憶領域に対して、前記ターゲットによりデータの書き込みあるいは読み出しを行わせてデータを交換する通信制御方法であって、

前記イニシエータは、ターゲットに対して発行した命令を、それぞれの命令について処理を完了した旨の処理完了通知をターゲットから受けるまで、再発行可能な状態で発行済の命令を保持することを特徴とする通信制御方法。

【請求項2】 前記イニシエータとターゲットとの間の接続が一旦切断されて再接続された場合、前記イニシエータは、ターゲットに対して発行した命令と、その命令により書き込みあるいは読み出しが行われる記憶領域との対応づけを保持することを特徴とする請求項1に記載の通信制御方法。

【請求項3】 前記イニシエータとターゲットとの間の接続が一旦切断されて再接続された場合、前記イニシエータは、ターゲットに対して発行した命令により書き込みあるいは読み出しが行われる記憶領域の内容を保持することを特徴とする請求項1に記載の通信制御方法。

【請求項4】 前記イニシエータとターゲットとの間の接続が一旦切断されて再接続された場合、前記イニシエータは、再発行可能な状態で保持されている命令を、ターゲットに対して再発行することを特徴とする請求項1に記載の通信制御方法。

【請求項5】 前記ターゲットは、イニシエータから発行された命令の種類ごとに、実行中の命令の識別子と、実行中の命令に対して書き込みあるいは読み込みを行っている記憶領域のオフセットを保持することを特徴とする請求項1に記載の通信制御方法。

【請求項6】 前記イニシエータとターゲットとの間の接続が一旦切断されて再接続された場合、前記ターゲットは、イニシエータから発行された命令の識別子と保持されている識別子とを比較し、一致している場合には、その命令にしたがって、保持されている記憶領域のオフセットから読み出しあるいは書き込みを行い、一致していない場合には、イニシエータに当該命令に対する処理完了通知を、イニシエータの記憶領域に対する読み出しあるいは書き込みを行わずに発行することを特徴とする請求項1に記載の通信制御方法。

【請求項7】 イニシエータからターゲットに対して命令を発行することで、前記イニシエータの有する記憶領域に対して、前記ターゲットによりデータの書き込みあるいは読み出しを行わせてデータを交換する通信システムであって、

前記イニシエータは、ターゲットに対して発行した命令を、それぞれの命令について処理を完了した旨の処理完了通知をターゲットから受けるまで、再発行可能な状態

で発行済の命令を保持することを特徴とする通信システム。

【請求項8】 前記イニシエータとターゲットとの間の接続が一旦切断されて再接続された場合、前記イニシエータは、ターゲットに対して発行した命令と、その命令により書き込みあるいは読み出しが行われる記憶領域との対応づけを保持することを特徴とする請求項7に記載の通信システム。

【請求項9】 前記イニシエータとターゲットとの間の接続が一旦切断されて再接続された場合、前記イニシエータは、ターゲットに対して発行した命令により書き込みあるいは読み出しが行われる記憶領域の内容を保持することを特徴とする請求項7に記載の通信システム。

【請求項10】 前記イニシエータとターゲットとの間の接続が一旦切断されて再接続された場合、前記イニシエータは、再発行可能な状態で保持されている命令を、ターゲットに対して再発行することを特徴とする請求項7に記載の通信システム。

【請求項11】 前記ターゲットは、イニシエータから発行された命令の種類ごとに、実行中の命令の識別子と、実行中の命令に対して書き込みあるいは読み込みを行っている記憶領域のオフセットを保持することを特徴とする請求項7に記載の通信システム。

【請求項12】 前記イニシエータとターゲットとの間の接続が一旦切断されて再接続された場合、前記ターゲットは、イニシエータから発行された命令の識別子と保持されている識別子とを比較し、一致している場合には、その命令にしたがって、保持されている記憶領域のオフセットから読み出しあるいは書き込みを行い、一致していない場合には、イニシエータに当該命令に対する処理完了通知を、イニシエータの記憶領域に対する読み出しあるいは書き込みを行わずに発行することを特徴とする請求項7に記載の通信システム。

【請求項13】 接続されたターゲットに対して命令を発行することで、ターゲットによりデータの書き込みあるいは読み出しを行わせてデータを交換する通信制御装置であって、

ターゲットにより読み書きさせる記憶領域を含むメモリと、

ターゲットに対して発行した命令を、それぞれの命令について処理を完了した旨の処理完了通知をターゲットから受けるまで、再発行可能な状態で発行済の命令を保持する保持手段とを備えることを特徴とする通信制御装置。

【請求項14】 前記ターゲットに対して発行した命令と、当該命令に基づいてターゲットにより読み書きさせる前記記憶領域との対応づけを保持する保持手段を更に備えることを特徴とする請求項13に記載の通信制御装置。

【請求項15】 前記ターゲットとの間の接続が一旦切

10

20

30

40

50

断されて再接続された場合、前記保持手段により再発行可能な状態で保持されている命令を、ターゲットに対して再発行することを特徴とする請求項13に記載の通信制御装置。

【請求項16】 接続されたイニシエータから発行される命令に基づいてイニシエータの記憶領域に対してデータの書き込みあるいは読み出しを行ってデータを交換する通信制御装置であって、
前記イニシエータから発行された命令の種類ごとに、実行中の命令の識別子を保持する命令保持手段と、
実行中の命令に対して書き込みあるいは読み込みを行っている記憶領域のオフセットを保持するオフセット保持手段とを備えることを特徴とする通信制御装置。

【請求項17】 前記イニシエータとの間の接続が一旦切断されて再接続された場合、前記イニシエータから発行された命令の識別子と前記命令保持手段により保持されている識別子とを比較し、一致している場合には、その命令にしたがって、前記オフセット保持手段により保持されている記憶領域のオフセットから読み出しあるいは書き込みを行い、一致していない場合には、イニシエータに当該命令に対する処理完了通知を、イニシエータの記憶領域に対する読み出しあるいは書き込みを行わずに発行することを特徴とする請求項16に記載の通信制御装置。

【請求項18】 コンピュータにより、それに接続されたターゲットに対して命令を発行することで、ターゲットによりデータの書き込みあるいは読み出しを行わせてデータを交換するためのプログラムを格納するコンピュータ可読記憶媒体であって、前記プログラムにより、
ターゲットに対して発行した命令を、それぞれの命令について処理を完了した旨の処理完了通知をターゲットから受けるまで、再発行可能な状態で発行済の命令を保持する保持手段と、

前記ターゲットに対して発行した命令と、当該命令に基づいてターゲットにより読み書きさせる記憶領域との対応づけを保持する保持手段と、

前記ターゲットとの間の接続が一旦切断されて再接続された場合、前記保持手段により再発行可能な状態で保持されている命令を、ターゲットに対して再発行する手段とを実現することを特徴とする記憶媒体。

【請求項19】 コンピュータにより、それに接続されたイニシエータから発行される命令に基づいてイニシエータの記憶領域に対してデータの書き込みあるいは読み出しを行ってデータを交換するためのプログラムを格納するコンピュータ可読の記憶媒体であって、前記プログラムにより、

前記イニシエータから発行された命令の種類ごとに、実行中の命令の識別子を保持する命令保持手段と、
実行中の命令に対して書き込みあるいは読み込みを行っている記憶領域のオフセットを保持するオフセット保持

手段と、

前記イニシエータとの間の接続が一旦切断されて再接続された場合、前記イニシエータから発行された命令の識別子と前記命令保持手段により保持されている識別子とを比較し、一致している場合には、その命令にしたがって、前記オフセット保持手段により保持されている記憶領域のオフセットから読み出しあるいは書き込みを行い、一致していない場合には、イニシエータに当該命令に対する処理完了通知を、イニシエータの記憶領域に対する読み出しあるいは書き込みを行わずに発行する手段とを実現することを特徴とする記憶媒体。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、例えばホストコンピュータとプリンタ等の機器同士を接続する通信制御方法及び装置に関する。

【0002】

【従来の技術】近年、IEEE1394インターフェースが、コンピュータと周辺機器、あるいは周辺機器同士の接続に用いられつつある。IEEE1394インターフェースは、セントロニクスインターフェースなどのハンドシェイク方式に比べて高速で、しかも双方向通信が可能である。また、メモリバスモデルのインターフェースであり、IEEE1394インターフェースで接続された機器は、相手の機器に対して指定されたアドレスのデータを読み書きすることができる。

【0003】このIEEE1394は、広範囲に応用するための物理層及びリンク層のプロトコルを定めたもので、機器ごとの詳細なプロトコルは定められていない。そのため、物理・リンク層としてIEEE1394を利用したSBP(Serial Bus Protocol)-2などのトランスポート層のプロトコルが提案されている。トランスポート層はアプリケーションに対してデータ転送機能を提供する層であり、この層を利用するアプリケーションは、互いにデータの交換が可能となる。

【0004】このSBP-2なるプロトコルは、IEEE1394のメモリバスモデルとしての特徴を活かしたプロトコルであり、データの受信側がそれ自身の都合に応じてデータを受信することができる。これに対して、SBP-2以外のプロトコルには、非同期に発生するデータの転送が可能となるものも、マルチチャネルが実現できるものもあるが、そのようなプロトコルではIEEE1394のメモリバスモデルとしての特徴を活かすことができない。すなわち、ホストとプリンタの通信であれば、プリンタ側の都合に応じてデータ転送を行うことができず、ホストがプリンタの状態を監視しながらデータ転送を行わねばならないことになる。

【0005】SBP-2では、データを転送する場合に、まず送信側でログインという操作を行って通信相手との間のチャネルを確立する。この際、ログインした側

がイニシエータと呼ばれ、イニシエータと接続された相手側がターゲットと呼ばれる。データ転送は、イニシエータからの指示に応じて、ターゲットからイニシエータのバッファにデータを読み書きすることで行われる。この方式においては、イニシエータは、送信するデータを格納したバッファのアドレスやサイズ等が書かれたORB (Operation Request Block)を作成し、そのORBのアドレスをターゲットに知らせる。ターゲットでは自分の都合に合わせてORBに書かれたアドレスやサイズに基づいてイニシエータからデータを読み出し、あるいはデータを書き込み、それらの処理後ステータスブロックを作成して、処理の状態をイニシエータに知らせることになる。

【0006】

【発明が解決しようとする課題】このIEEE1394の上に構築されるSBP-2プロトコルを利用して通信を行う場合、特に、ホストコンピュータ等のデータ源をイニシエータとし、それからターゲットであるプリンタ装置などの周辺機器へのデータ転送に適用する場合、次のような4つの問題点があった。(問題点1)全2重通信のために手順が複雑である。

【0007】SBP-2では、基本的にデータの転送はイニシエータによって管理され、ターゲットはイニシエータへの非同期のデータ転送を行うことができない。すなわち、SBP-2では、ターゲットがイニシエータへのデータ転送を行いたい時には、イニシエータに対してアンソリシテドステータスでデータリード要求を送り、イニシエータはそれに応じてORB生成し、ペンディング中のORB (イニシエータからターゲットへのデータ転送要求などが含まれている)のリストの末尾にその生成したORBをつける。このORBは先頭から順番に処理されるため、ターゲットがイニシエータにリード要求を発行したタイミングではなく、イニシエータ側のORBの処理が進み、ターゲットからのデータリード要求によるORBが処理されたときに初めてターゲットからイニシエータにデータが転送されることになる。すなわち、双方向の非同期なデータ転送が行えず、ターゲットからイニシエータに転送すべきデータが非同期に発生する場合、例えばターゲットがプリンタであれば、そのプリンタでエラーが発生する場合など、直ちにイニシエータに伝達すべきデータを即時に伝達できない。

【0008】このため、例えばプリンタで非同期に発生するデータを即座にホストに転送するためには、プリンタをイニシエータとしてログイン手続きを行い、ホストコンピュータをターゲットとするデータ転送を行わねばならない。

【0009】このようにホストコンピュータとプリンタで互いにログインしてそれぞれがイニシエータでありターゲットであるような状況では、ホストコンピュータ及びプリンタの双方にイニシエータとしてのプロセスとタ

ーゲットとしてのプロセスを備えなければならない。また、ログインの操作もプリンタから行わねばならない。

【0010】プリンタのような画像を扱う周辺装置では、画像処理のために大量のメモリ資源やプロセッサ資源を消費する。そのために、装置の構成を簡略化して原価を節減したり、処理を迅速に行うため画像処理用途以外に用いられる資源をできるだけ節約しなければならない。しかし、上述のように多くのプロセスを稼働することになればその分多くの資源を消費することになり、原価低減・処理の効率化という目的に反することになる。

【0011】また、ホストコンピュータとプリンタとの関係であれば、各方向に流れるデータは、印刷データとそれに対する処理状況といったように、互いに関連付けられるものであるが、各方向について独立したログインによりチャンネルを設定すると、それらデータとレスポンスとを関連づけなければならない、そのための処理手順を新たに追加する必要がある。

【0012】このようにIEEE1394及びSBP-2をそのままホストコンピュータ・プリンタ装置間の通信に適用することは適切でなく、各装置において必要とされる資源を減らすことや効率を向上させることが難しかった。(問題点2)マルチチャンネルを実現できない。

【0013】最近、周辺装置として種々の機能を複合させた複合機が利用されつつある。例えば、ファクシミリ装置を、スキャナ単体、プリンタ単体、ファクシミリとしてホストコンピュータ等から利用できるデジタル複合機などがある。このような装置を利用する際には、各単体機能ごとに独立した複数のチャンネルを介して通信を行えば、同時に複数の機能を利用することができる。

【0014】しかしながら、SBP-2では、マルチチャンネルを提供できないため、そのように単体機能を同時に利用することが難しい。(問題点3)バスリセットに対応できない。

【0015】IEEE1394では、1394シリアルバスへの機器の新たな接続や取り外し、あるいは、接続された機器の電源投入や切断といった、ネットワーク構成の変化要因となる状態変化が発生するとバスリセットが発生する。バスリセットは、バス上で上記のような状態変化を検知したノードがバスリセット信号をバス上に送信することで発生する。発生されたバスリセット信号は、ノードからノードへと伝達され、ネットワーク上の全ノードがバスリセット信号を受信すると、バスリセットのための一連の動作が各ノードで行われる。

【0016】このようにバスリセットは、ネットワーク上のノードにおける処理とは非同期に発生する。また、バスリセットの原因となったノードとアプリケーションについては無関係なノードであっても、一旦バスリセットが発生してしまえばバスリセットの処理を行わねばならない。このバスリセットの過程で、SBP-2で通信を行っているノードでは、設定されていたコネクション

が切断されてしまい、再度コネクションをつなげてもバスリセット直前の状態から処理を続行できる保証が与えられていない。

【0017】本発明は上記従来例に鑑みてなされたもので、1回のログインで全2重通信（互いに非同期な双方向通信）を可能とし、また、データの交換に必要なプロセスやメモリといった資源を効率的に利用できる通信制御方法及び装置及びそれを用いた印刷装置を提供することを目的とする。

【0018】また、マルチチャネルを実現する通信制御方法及び装置及びそれを用いた印刷装置を提供することを目的とする。

【0019】また、バスリセットが発生しても、バスリセット直前の状態から処理の続行を保証する通信制御方法及び装置及びそれを用いた印刷装置を提供することを目的とする。

【0020】

【課題を解決するための手段】上記目的を達成するために本発明は次のような構成から成る。すなわち、イニシエータからターゲットに対して命令を発行することで、前記イニシエータの有する記憶領域に対して、前記ターゲットによりデータの書き込みあるいは読み出しを行わせてデータを交換する通信制御方法であって、前記イニシエータは、ターゲットに対して発行した命令を、それぞれの命令について処理を完了した旨の処理完了通知をターゲットから受けるまで、再発行可能な状態で発行済の命令を保持する。

【0021】あるいは、イニシエータからターゲットに対して命令を発行することで、前記イニシエータの有する記憶領域に対して、前記ターゲットによりデータの書き込みあるいは読み出しを行わせてデータを交換する通信システムであって、前記イニシエータは、ターゲットに対して発行した命令を、それぞれの命令について処理を完了した旨の処理完了通知をターゲットから受けるまで、再発行可能な状態で発行済の命令を保持する。

【0022】あるいは、接続されたターゲットに対して命令を発行することで、ターゲットによりデータの書き込みあるいは読み出しを行わせてデータを交換する通信制御装置であって、ターゲットにより読み書きさせる記憶領域を含むメモリと、ターゲットに対して発行した命令を、それぞれの命令について処理を完了した旨の処理完了通知をターゲットから受けるまで、再発行可能な状態で発行済の命令を保持する保持手段とを備える。

【0023】あるいは、接続されたイニシエータから発行される命令に基づいてイニシエータの記憶領域に対してデータの書き込みあるいは読み出しを行ってデータを交換する通信制御装置であって、前記イニシエータから発行された命令の種類ごとに、実行中の命令の識別子を保持する命令保持手段と、実行中の命令に対して書き込みあるいは読み込みを行っている記憶領域のアドレスを

保持するアドレス保持手段とを備える。

【0024】あるいは、コンピュータにより、それに接続されたターゲットに対して命令を発行することで、ターゲットによりデータの書き込みあるいは読み出しを行わせてデータを交換するためのプログラムを格納するコンピュータ可読記憶媒体であって、前記プログラムにより、ターゲットに対して発行した命令を、それぞれの命令について処理を完了した旨の処理完了通知をターゲットから受けるまで、再発行可能な状態で発行済の命令を保持する保持手段と、前記ターゲットに対して発行した命令と、当該命令に基づいてターゲットにより読み書きさせる記憶領域との対応づけを保持する保持手段と、前記ターゲットとの間の接続が一旦切断されて再接続された場合、前記保持手段により再発行可能な状態で保持されている命令を、ターゲットに対して再発行する手段とを実現する。

【0025】あるいは、コンピュータにより、それに接続されたイニシエータから発行される命令に基づいてイニシエータの記憶領域に対してデータの書き込みあるいは読み出しを行ってデータを交換するためのプログラムを格納するコンピュータ可読の記憶媒体であって、前記プログラムにより、前記イニシエータから発行された命令の種類ごとに、実行中の命令の識別子を保持する命令保持手段と、実行中の命令に対して書き込みあるいは読み込みを行っている記憶領域のアドレスを保持するアドレス保持手段と、前記イニシエータとの間の接続が一旦切断されて再接続された場合、前記イニシエータから発行された命令の識別子と前記命令保持手段により保持されている識別子とを比較し、一致している場合には、その命令にしたがって、前記アドレス保持手段により保持されている記憶領域のアドレスから読み出しあるいは書き込みを行い、一致していない場合には、イニシエータに当該命令に対する処理完了通知を発行する手段とを実現する。

【0026】

【発明の実施の形態】〔第1の実施の形態〕本発明の第1の実施の形態として、ホストコンピュータとプリンタとを、IEEE1394で接続し、その上に構築されたSBP-2プロトコルを用いた、本発明に係るプロトコル（以下SHPTと呼ぶ）に従ってデータ転送を行うプリンタシステムを説明する。図24は、そのプリンタシステムにおけるハードウェア構成図である。

＜システムのハードウェア構成＞図24において、ホストコンピュータ200は、ROM3のプログラム用ROMに記憶された文書処理プログラム等に基づいて、図形、イメージ、文字、表（表計算等を含む）等が混在した文書処理を実行するCPU1を備える。CPU1はシステムデバイス4に接続される各デバイスをCPU1が総括的に制御する。また、ROM3のプログラム用ROMには、CPU1の制御プログラム等が記憶され、RO

M3のフォント用ROMには上記文書処理の際に使用するフォントデータ等が記憶され、ROM3のデータ用ROMは上記文書処理等を行う際に使用する各種データが記憶される。RAM2は、CPU1の主メモリ、ワークエリア等として機能する。プログラムはRAM2に格納されても良い。また、RAM2には、送信データバッファやORBを格納するシステムメモリが確保される。

【0027】キーボードコントローラ(KBC)5は、キーボード9や不図示のポインティングデバイスからのキー入力を制御する。CRTコントローラ(CRTC)6は、CRTディスプレイ(CRT)10の表示を制御する。メモリコントローラ(MC)7は、ブートプログラム、種々のアプリケーション、フォントデータ、ユーザファイル、編集ファイル等を記憶するハードディスク(HD)、フロッピーディスク(FD)等の外部メモリ11とのアクセスを制御する。1394インターフェース8は、IEEE1394規格に従ってプリンタ100に接続されて、プリンタ100との通信制御処理を実行する。なお、CPU1は、例えばRAM2上に設定された表示情報RAMへのアウトラインフォントの展開(ラスタライズ)処理を実行し、CRT10上でのWYSIWYGを可能としている。また、CPU1は、CRT10上の不図示のマウスカーソル等で指示されたコマンドに基づいて登録された種々のウィンドウを開き、種々のデータ処理を実行する。

【0028】プリンタ100において、プリンタCPU12は、ROM13のプログラム用ROMに記憶された制御プログラム等或いは外部メモリ14に記憶された制御プログラム等に基づいてシステムバス15に接続される各種のデバイスとのアクセスを総括的に制御し、印刷部インタフェース16を介して接続される印刷部(プリンタエンジン)17に出力情報としての画像信号を出力する。また、このROM13のプログラムROMには、後述する各種エージェントや画像処理を実現するCPU12の制御プログラム等が記憶される。ROM13のフォント用ROMには上記出力情報を生成する際に使用するフォントデータ等が記憶され、ROM13のデータ用ROMには、ハードディスク等の外部メモリ14が無いプリンタの場合には、ホストコンピュータ上で利用される情報等が記憶されている。CPU12は1394インターフェース18を介してホストコンピュータとの通信処理が可能となっており、プリンタ内の情報等をホストコンピュータ200に通知可能に構成されている。

【0029】RAM19は、CPU12の主メモリ、ワークエリア等として機能するRAMで、図示しない増設ポートに接続されるオプションRAMによりメモリ容量を拡張することができるように構成されている。なお、RAM19は、出力情報展開領域、環境データ格納領域、NVRAM等に用いられる。

【0030】前述したハードディスク(HD)、ICカ

ード等の外部メモリ14は、メモリコントローラ(MC)20によりアクセスを制御される。外部メモリ14は、オプションとして接続され、フォントデータ、エミュレーションプログラム、フォームデータ等を記憶する。また、操作パネル18には、操作のためのスイッチおよびLED表示器等が配されている。また、前述した外部メモリは1個に限らず、少なくとも1個以上備え、内蔵フォントに加えてオプションフォントカード、言語系の異なるプリンタ制御言語を解釈するプログラムを格納した外部メモリを複数接続できるように構成されていても良い。さらに、図示しないNVRAMを有し、操作パネル1012からのプリンタモード設定情報を記憶するようにしても良い。

【0031】<イニシエータの構成>上述したハードウェア構成において、プリンタ100をターゲットとし、ホストコンピュータ200をイニシエータとした通信システムを、図1及び図2に示す。本実施形態に置いては、これら構成は、それぞれホストコンピュータ及びプリンタにおけるCPUによりプログラムを実行することで実現される。まず、図2のイニシエータから説明する。

【0032】図2において、イニシエータであるホストコンピュータにおいては、プリンタドライバ等のクライアント206はSHPT処理部201を介してプリンタに対するデータ転送要求を発行するとともに、プリンタからの応答を受ける。

【0033】SHPT処理部201は、システムメモリに生成されるI/O要求キューやORBを管理している。リードI/O要求キュー203にはクライアント206からのリード要求コマンドが、ライトI/O要求キュー202にはクライアント206からのライト要求コマンドが、それぞれキューイングされる。ORBはイニシエータであるホストコンピュータからターゲットであるプリンタへ、あるいはプリンタからホストコンピュータへ渡されるデータバッファのアドレスやサイズ等を格納したブロックであり、先頭のORBからORBリスト204に順次リンクされている。このORBについては、次のような処理の原則がある。

(1) リードI/O要求キュー及びライトI/O要求キューそれぞれの先頭から取り出したコマンドから順にORBが作成され、ORBリストの末尾につけられる。リード/ライトの順序には特に制約はない。

(2) ORBは、ORBリストの先頭から順次フェッチされる。完了通知(ステータスブロック)を受信すると、そのステータスに対応するORBはORBリストからはずされる。

(3) リンクされるORBの数の上限は、後述するターゲットにおけるリードプリフェッチキュー及びライトプリフェッチキューの容量(それぞれ処理中のコマンドも含む)の合計と同一である。

10

20

30

40

50

(1)、(2)項により、ライトI/O要求、リードI/O要求は、それぞれのI/O要求において、発生した順にターゲットに発行されることが保証される。また、

(3)項により、ORBリスト中のI/O要求はターゲットに渡されていることが保証される。この(3)項を実現するために、SHPT処理部201は、プリンタの2つのプリフェッチキューそれぞれについて2つのカウンタを用意している。ひとつはCurrentReadQEというカウンタ203aであり、後述するターゲットのリードプリフェッチキューの現在の空容量を示している。また、

もうひとつはCurrentWriteQEというカウンタ202aであり、ターゲットのライトプリフェッチキューの空き容量を示している。これらカウンタの初期値はターゲットとなる機器が保持しており、ログイン時等にターゲットから読み出され、インシエータに記憶される。これらカウンタはORBの生成と削除に合わせて増減される。
【0034】また、ORBを生成した際には、ターゲット側のドアベルレジスタと呼ばれるレジスタに所定値を書き込む、あるいは、リストの先頭のORBのアドレスをORBポインタと呼ばれるレジスタに書き込むことにより、ORBが発生したことをターゲットに知らせる。既にターゲットが実行しているORBリストの最後にORBした場合にはドアベルレジスタへの書き込みを行い、ORBリストがない状態で新規に先頭のORBを作成した場合には、ORBポインタへの書き込みを行う。この手続きはSBP-2で規定されている。

【0035】また、SHPT処理部201は、ステータスFIFO205を含む。1394インターフェース109を介して受信されたステータスは、SHPT処理部201により処理される。SHPT処理部201は、受信したステータスに対応するORBをORBリスト204から外すとともに、そのORBに対応するコマンドをI/O要求キューから外す。

【0036】インシエータたるホストコンピュータは以上のような機能的な構成からなっている。

【0037】<ターゲットの構成>図1は、ターゲットたるプリンタの機能上の構成を示すブロック図である。図1において、ドアベルレジスタ101aは、インシエータによって書き込まれるレジスタであり、ORBが新たに生成されたことを示す。また、ORBポインタ101bには、新たに生成されたORBのアドレスがインシエータにより書き込まれる。フェッチエージェント101は、ドアベル101aに書き込みがあると、1394インターフェース101を介してORBポインタで示されるORBを読み込む。ORBディスパッチャ102は、フェッチエージェント101により読み込まれたORBがライトコマンドORBであればライトコマンドをライトプリフェッチキュー103の末尾につけ、リードコマンドORBであればリードコマンドをリードプリフェッチキュー105の末尾につける。このコマンドの種

類は、ORBのファンクションフィールドの値に応じて判定される。

【0038】ライト実行エージェント104及びリード実行エージェント106は、それぞれライトプリフェッチキュー103及びリードプリフェッチキュー105にキューイングされたコマンドを先頭から取り出し、インシエータのバッファへのデータの書き込み及びインシエータのバッファからのデータの読み出しを行う。その後、ホストコンピュータにノーマルステータスを返す。

【0039】なお、リード実行エージェント106は、例えばPDLを解釈実行してラスタデータを生成するラスタライザ等のクライアント108からのデータ転送要求に応じて、リードプリフェッチキュー内のリードコマンドで指定されたインシエータのバッファにデータを書き込む。このデータ転送要求は、インシエータからのライトコマンドやリードコマンドとは非同期に発生するため、インシエータはリードコマンドORBによってリードプリフェッチキューにリードコマンドを常時キューイングしておく。ターゲットは、リードコマンドがプリフェッチキューにキューイングされている限り、いつでもデータをインシエータに渡すことができる。なお、バスインターフェース110は、ターゲットであるプリンタ100からインシエータであるホストコンピュータ200のシステムメモリ208の所望のメモリロケーションにアクセスするためのインターフェースである。

【0040】以上、インシエータ及びターゲットの構成と動作を簡単に説明した。これらを更に詳しく説明する前に、まずORBの詳しい内容を説明する。

【0041】<コマンドORB (Operation Request Block)の内容>図7はORB一般の構成を示す図である。図7(A)において、"Next_ORB" (リンク) フィールド301は、次のORBへのリンクである。次のORBがない場合にはそれを表す所定の値が入れられる。なお、先頭のORBは、所定のアドレスレジスタにより示される。"data_descriptor" (データデスク립タ) フィールド302は、データバッファのアドレスを示す。"d" (方向) フィールド303は、ホストコンピュータからプリンタへのデータ転送(0:ライト)か、プリンタからホストコンピュータへのデータ転送(1:リード)かを示す。"data_size" (データサイズ) フィールド304は、アドレスフィールド302で指し示されるデータバッファのサイズを示す。以上、図の点線よりも上のフィールド(未説明のフィールドも含めて)がSBP-2で規定されているフィールドであり、これから説明するフィールド305~308が、SHPTに固有の処理に用いられるフィールドである。

【0042】"PSID"フィールド305a、"SSID"フィールド305bは、使用されるチャンネルIDを表わしている。"function" (ファンクション) フィールド306は、図7(B)に示すようにORBの種類を示してい

る。0Hがライトコマンド、40Hがリードコマンドを表わす。

【0043】"t" (タグ) ビット307はデータタグを表している。"seq_ID" (シーケンスID) フィールド308は、ファンクションごと(リードあるいはライトコマンド)に、ORBの生成順に振られるシーケンシャルな識別子である。この識別子は、本実施形態ではORBが生成されるごとに1ずつ増えるように与えられる。なお、このシーケンスIDフィールドに値を与える基準となるカウンタは、バスリセットや他の障害による影響を受けない領域に設けられる。それは、ターゲットにおいては、シーケンスIDの連続性を前提として処理を行うためである。コントロールブロックフィールド309には、ファンクションフィールド306の値や、各ファンクションに対してターゲットで実現される機能に応じて様々な値が入れられる。なお、バッファのアドレスやサイズについては、I/O要求キューに入れられたコマンドからORB生成時に得る。次に、機能ごとにORBの内容を説明する。

【0044】(ライトコマンドORB) 図8はファンクション=0HのライトコマンドORBを示している。このコマンドは、指定したバッファ内のデータをイニシエータからターゲットに渡すためのコマンドである。ファンクションフィールドの値は0Hであり、シーケンスIDは、ひとつのチャンネルごとに、各ライトコマンドに対して昇順に与えられる。

【0045】(リードコマンドORB) 図9はファンクション=40HのリードコマンドORBを示している。このコマンドは、イニシエータがターゲットからデータを読み出すためのコマンドである。ファンクションフィールドの値は40Hであり、シーケンスIDは、ひとつのチャンネルごとに、各リードコマンドに対して昇順に与えられる。

【0046】(ステータスブロック) 図10(A)～(C)は、ターゲットからイニシエータに返されるステータスの形式及びその内容を示している。図10(A)において、点線よりも上のフィールドはSBP-2で規定されたフィールドであるため、特に説明をしない。PSIDフィールド401、SSIDフィールド402、ファンクションフィールド403、タグフィールド404は、図7のコマンドORBと同様のフィールドである。図10(B)にファンクションフィールドの値とその意味を示す。

【0047】"SHPT_status" (SHPTステータス) フィールド図10(C)に示すような意味を持つ。すなわち、値が0であれば、そのステータス対応するコマンドORBについての処理は完了したことを示す。例えば、対応するコマンドORBがライトコマンドであれば、SHPTステータスが0のステータスブロックは、対応するライトコマンドORBで差し示されたバッファから、

ターゲットがデータを全て読み出し、ターゲットの持つバッファに書き込み終えたことを意味する。また、値が1であれば、対応するコマンドORBの処理が部分的な終了していることを示す。また、値が2であれば、対応するコマンドORBの処理はエラーにより行われなかったことを示す。このエラーを受けたイニシエータは、クライアントにエラーである旨知らせるなどしてエラーに対応する。

【0048】"length_hi" (レングスハイ) フィールド405及び"length_lo" (レングスロウ) フィールド406は、処理されたデータ長を示す。

【0049】(ライトステータスブロック) 図11は、ファンクション=0のライトステータスブロックであり、ライトコマンドに対してターゲットからイニシエータに渡される。処理の状態は、SHPTステータスにより、図10(C)のように表わされる。

【0050】(リードステータスブロック) 図12は、ファンクション=40Hのリードステータスブロックであり、リードコマンドに対してターゲットからイニシエータに渡される。処理の状態は、SHPTステータスにより、図10(C)のように表わされる。

【0051】<ORBの管理>次に、以上説明したイニシエータ及びターゲットの構成や、コマンドORB及びステータスブロックの構成に基づいて、ORBがどのように用いられているか説明する。

【0052】図3は、イニシエータ及びクライアント間におけるコマンドの流れを示している。まずライトコマンドについて説明する。イニシエータのクライアントからデータ転送要求が発生すると、クライアントはターゲットに渡すデータを格納したバッファのアドレス及びサイズを含むライトコマンドを、ライトI/O要求キュー202の末尾につける。SHPT処理部201では、ターゲットのライトブリフエッチキュー103の空き容量を示すCurrentWriteQUE202aをチェックし、空きがあればそのライトコマンドからORBを生成してORBリスト204の末尾にリンクするが、空きがなければ空きを待つ。図3の状態では、ライトブリフエッチキューはライトコマンドW1～W4によっていっぱいであり、その後のライトコマンドW5及びW6はライトI/O要求キューで待たされている。なお、ORBが生成されてもそれに対応するライトコマンドはライトI/O要求キューから削除されず、ターゲットからの完了ステータスを受けた時点で削除される。

【0053】SHPT処理部201は、ORBをORBリストに入れると、1394インターフェースを介してターゲットのドアベル101aに書き込む、あるいは、ORBポインタ101bに新たなORBのアドレスを書き込む。ディスパッチャ102は、そのORBに基づいて、ライトコマンドをライトブリフエッチキュー103の末尾につける。ライト実行エージェント104は、ラ

イトプリフェッチキュー内のコマンドを順次実行する。すなわち、ターゲットのクライアントの用意したバッファに、ライトコマンドで示されるバッファの内容を書き込む。ライト実行エージェントは、コマンドの処理が完了すると、SHPTステータスが「完了」であるステータスブロックをイニシエータのステータスFIFO205に書き込む。

【0054】SHPT処理部201は、ステータスFIFOに入れられたステータスブロックを先頭から処理する。すなわち、SHPTステータスが「完了」であれば、そのステータスブロックに対応するORBをORBリストから外し、それとともにライトI/O要求キュー内の対応するライトコマンドを削除する。以上の手順は、リードコマンドについても全く同様である。ただし、リードコマンドは、ターゲットにおいてリードすべきデータが発生しない限り処理されない。従って、イニシエータから発行されたリードコマンドORBは、リードすべきデータが発生するまでリードプリフェッチキューにキューイングされたままということになる。

【0055】イニシエータがホストコンピュータ、ターゲットがプリンタである場合、ホストコンピュータからプリンタに印刷データを渡すためには、上述のライトコマンドを用いて、イニシエータのクライアントであるプリンタドライバから、ターゲットのクライアントであるラスタライザに印刷データが渡される。また、ホストコンピュータがプリンタの構成や状態を示す情報を要求する場合には、その旨のコマンド（クライアントレベルのコマンド）をプリンタにライトコマンドで渡す。それを受けたプリンタ（のクライアント）は、リードプリフェッチキューにキューイングされているリードコマンドを用いて、ホストコンピュータに要求されたデータを渡す。さらに、プリンタでエラーが発生したような場合には、プリンタのクライアントは、リードプリフェッチキューにキューイングされているリードコマンドを用いて、自発的にエラー情報をホストコンピュータに渡すことができる。このため、ホストコンピュータはプリンタと接続している間は、運用上少なくとも1つのリードORBをプリンタに対して発行しておく。さらに、リードコマンドを常時リードプリフェッチキューにキューイングさせておくためには、少なくとも2つのリードORBをプリンタに対して発行しておくことが望ましい。

【0056】図6はORBリストの更新の様子を示している。状態1から、ライトコマンドW1とリードコマンドR1とが処理されてそれらのORBはORBから削除される（状態2）。このため、ライトORB W2, W3のリンク先はなくなる。この場合、これらORBは既にターゲットに渡っているため、これらをORBリストにつなぎなおす必要はない。また、ORBリストになくとも、I/O要求キューには対応コマンドが残っているため、後述するエラー復旧処理も正常に行うことができ

る。

【0057】＜エラーの復旧（イニシエータ）＞既に説明したように、SBP-2では、ネットワークに異常（エラー）に起因するバスリセットにより、イニシエータとターゲットとの間の接続は断たれてしまう。そこで、SHPTでは、バスリセットにより失われた状態を復旧するための手順を定めている。図4～図6を参照してその手順の概要を説明する。

【0058】図4はイニシエータの状態を示している。図4において、状態1では、ライトコマンドW1～W4のライトORBと、リードコマンドR1～R3のリードORBが発行されており、ライトコマンドW5, W6はライトI/O要求キュー202にとどめられた状態にある。ターゲットにより1つのライトコマンドと2つのリードコマンドとが処理され、イニシエータは状態2に移る。処理されたコマンドに対応するORBはORBリスト204からはずされ、空き領域となっている。この状態では、ライトプリフェッチキューに空きが生じているが、まだライトコマンドW5に対応するORBは発行されていない。

【0059】ここでネットワーク上のエラーなどからバスリセットが発生する。バスリセットの処理が終了したなら、全てのORBをORBリストから消去する（状態3）。ライトI/O要求キュー202ならびにリードI/O要求キュー203は、バスリセットがあってもリセットされることはない。

【0060】イニシエータは、コネクションを再設定した後、ライトI/O要求キューならびにリードI/O要求キューを参照して、それらキューの内容から新たにORBを生成し、ターゲットに対して発行する。これが状態4である。ORBの生成時に対応するコマンドに対して何等かのマークをしておけば、バスリセット直前の状態を再現することができる。

【0061】また、図4には示されていないが、ステータスFIFOはバスリセットでは影響を受けないため、特に復旧する必要はない。

【0062】＜エラーの復旧（ターゲット）＞図5は、ターゲット側におけるバスリセット後の復旧の様子を示す図である。イニシエータはORBリストを復旧させるだけでその状態をバスリセット直前の状態に戻せるが、ターゲットはリード/ライトコマンドに応じてバッファに対してリード/ライトを行うため、リード/ライト中にバスリセットが発生した場合には、復旧後はその続きを行わねばならない。そのために、ターゲットのリード/ライト実行エージェントは、処理中のコマンドと、処理中のバッファの位置を記憶している。

【0063】図5では例としてリードコマンドの場合を示している。イニシエータがリードORBを発行し、ターゲットはそれをリード実行エージェントによって処理する。実行前に、リード実行エージェント106はこれ

から処理するリードORBのseq_IDフィールドを読み出し、シーケンス識別子(Sequenceidentifier)として記憶しておく。図5ではその値は「3」である。また、クライアントのバッファにデータを書き込むに当たり、現在書き込んでいるバッファのアドレスを実行ポインタ(Nextexepointer)として更新しつづける。図5ではその値は「offset」である。これらの領域はバスリセットによって消去されないメモリ領域に確保される。

【0064】バスリセットが終了したなら、イニシエータはORBを改めて発行してくるため、リード実行エージェントは、リードプリフェッチキューの先頭のORBのseq_IDフィールドが、記憶されているシーケンス識別子よりも古ければ、そのORBは既に処理は完了しているために、完了ステータスを返す。両者が一致すれば、Nextexepointerで示されるアドレスから書き込みを続行する。

【0065】ライトコマンドについても同じ要領で復旧される。

【0066】以上、本実施形態の印刷システムにおいて用いられるコマンドおよびステータスについて説明した。次に、イニシエータおよびターゲットにおけるコマンド及びステータスの処理手順を説明する。

【0067】＜イニシエータのクライアントによるデータ転送要求＞図13は、イニシエータのクライアントであるプリンタドライバ等から、ターゲットに対して、データを送信あるいは要求する際の手順を示している。

【0068】データ転送事項が発生したなら(ステップS1301)、そのために必要なコマンドがライトであるかリードであるか判定し(ステップS1302)、ライトならばそのためのデータ転送バッファがあるか調べ(ステップS1303)、また、ターゲットに渡すべき送信データの準備ができていないか調べて(ステップS1304)、すべて準備できていれば、バッファのアドレスやサイズ等、必要な引き数を与えてライトコマンドを作成し(ステップS1305)、ライト関数をコールする(ステップS1306)。

【0069】一方、リードコマンドであると判定されれば、データを受けるための転送バッファがあるか調べ(ステップS1307)、あればバッファのアドレスやサイズ等、必要な引き数を与えてリードコマンドを作成し(ステップS1308)、リード関数をコールする(ステップS1309)。

【0070】例えば、データ転送事項がプリンタに対する印刷データの送信ならば、PDLなど、クライアントレベルのコマンドやデータをバッファに用意し、ライトコマンドを作成することになるし、データ転送事項がプリンタからの状態の読み出しであれば、その旨を示すクライアントレベルのコマンドを用意してライトコマンドを作成するとともに、ターゲットからデータを受けるためのリードコマンドを作成する必要がある。また、ター

ゲットとの一連のデータ交換を行うに先立って、イニシエータのクライアントはリードコマンドを幾つか発行する。

【0071】図14は、イニシエータのクライアントが下位すなわちSHTPの層から処理終了通知を受けた際の処理手順である。まず、終了した処理がリードかライトか判定し(ステップS1401)、ライトであれば使用したデータバッファを解放する(ステップS1402)。一方、リードであれば、処理の終了はデータの受信の完了であるから、受信したデータに対応した処理を行い(ステップS1403)、データバッファを解放する(ステップS1404)。

【0072】＜イニシエータのSHPT処理部による処理＞図15～図18は、イニシエータのSHPT処理部による処理手順である。図15(A)及び(B)は、それぞれクライアントがコールするライト関数及びリード関数の内容を示す。ライト関数では、ライトI/O要求キューにライトコマンドを追加し、リード関数では、リードI/O要求キューにリードコマンドを追加する。

【0073】(I/O要求キュー管理)図16(A)、(B)は、それぞれライトI/O要求キュー管理手順、リードI/O要求キュー管理手順を示す。これらの手順は、それぞれ単独のスレッドで行われる。両者の処理内容は、管理対象がライトI/OキューとリードI/Oキューとで異なるものの、同一であるため、ライトI/O要求キュー管理手順のみを説明する。その説明のライトI/O要求キューをリードI/O要求キューに、カウンタCurrentWriteQUEをCurrentReadQUEに、参照番号のS160XをS161Xに読み替えれば、リードI/O要求キューの管理手順となる。

【0074】まず、ライトI/O要求キュー内にORB化されていないコマンドがあるか調べる(ステップS1601)。あれば、ターゲットのプリフェッチキューの空き容量を示すカウンタCurrentWriteQUEが0より大きい、すなわち空きがあるか判定する(ステップS1602)。空きがあればCurrentWriteQUEの値から1減じ(ステップS1603)、ライトI/O要求キューに格納されている、ORB化されていない先頭のコマンドを基に、シーケンスID(Seq_ID)やSHPTコマンド(Function)を付加して、ORBを作成する(ステップS1604)。

【0075】こうして作成したORBをORBリストにつけ(ステップS1605)、ターゲットのドアベルに書き込む、あるいは、ORBポインタにORBのアドレスを書き込む(ステップS1606)。このようにして、I/O要求キュー内のコマンドからORBが作成される。

【0076】(ステータスブロックに対する処理)図17は、ステータスブロックをターゲットから受けた際の処理手順である。ORBに対する処理完了のステータス

ブロックを受けると、対応するORBをORBリストから削除し（ステップS1701）、対応ORBがライトかそれともリードか判定する（ステップS1702）。ライトであれば、ライトI/O要求キューから、削除したORBの元となったコマンドを削除する（ステップS1703）。そして、ターゲットで処理を終えた分だけプリフェッチキューが空いたはずなので、カウンタCurrentWriteQUEに1加算する（ステップS1704）。最後に、クライアントに処理の終了を通知する（ステップS1705）。クライアントはこれを受けて、図14の処理を行うことになる。

【0077】一方、対応ORBがリードであると判定された場合には、リードI/O要求キューから、削除したORBの元となったコマンドを削除する（ステップS1706）。そして、ターゲットで処理を終えた分だけプリフェッチキューが空いたはずなので、カウンタCurrentReadQUEに1加算する（ステップS1707）。最後に、クライアントに処理の終了を通知する（ステップS1708）。

【0078】（エラー復旧処理）SHPT処理部における通常の処理は以上のとおりである。次にバスリセット後などに復旧する手順を図18により説明する。バスリセットが生じてターゲットとのコネクションが切断されてしまった場合、あらためてコネクションの再設定から行わねばならない。

【0079】まず、新規のORBのリンク処理を中断し（ステップS1801）、ORBリストを破棄する（ステップS1802）。この後コネクションの再接続命令を出して（ステップS1803）、コネクションの確立を調べ（ステップS1804）、コネクションが再度張られたなら、ライトI/O要求キュー及びリードI/O要求キュー内の先頭のコマンドから順に対応するORBを作成する（ステップS1805）。作成されたORBはORBリストにリンクされ（ステップS1806）、ORBポインタにアドレスを書き込んで（ステップS1807）、ターゲットにORBの発生を知らせる。

【0080】一方、コネクションが再設定できないまま一定時間経過すると（ステップS1808）、ライトI/O要求キュー及びリードI/O要求キュー内のコマンドをすべて破棄し（ステップS1809）、異常である旨をクライアントに知らせて（ステップS1810）、処理を終える。

【0081】異常の手順により、リセット後に、あらためてターゲットとのコネクションを確立し、リセット直前のORBリストを復旧することができる。

【0082】＜ターゲットによる処理＞図19～21は、ORBを受けたターゲットによる処理手順である。

【0083】（フェッチエージェントによる処理）フェッチエージェントはドアベルレジスタあるいはORBポインタに書き込みがあるとORBをフェッチする。ま

ず、ドアベルへの書き込みにより起動されたか判定し（ステップS1901）、そうでなく、ORBポインタへの書き込みにより起動されたのであればステップS1905に進む。一方、ドアベルへの書き込みがあったならば、フラグNullflagをテストする（ステップS1902）。このフラグは、ORBポインタに有効な値がセットされていないことを示すフラグであり、Nullflagがオフであれば、ORBポインタには処理すべきORBのアドレスが書き込まれているから、ステップS1905へと進む。Nullflagがオンであれば、現在のORBポインタで指されているORBのNext_ORBフィールドの値を、ORBポインタへと書き込み（ステップS1903）、Nullflagをオフにセットする（ステップS1904）。

【0084】こうしてORBポインタに処理すべきORBのアドレスが得られたなら、そのアドレスからORBを読み出す（ステップS1905）。読み出したORBのファンクションフィールドを参照して、そのSHPTコマンドがライトであるかリードであるか判定する（ステップS1906）。ライトであればライトORBをライトプリフェッチキューにつけ（ステップS1907）、リードであればリードORBをリードプリフェッチキューにつける（ステップS1908）。

【0085】この後、処理を終えたORBのNext_ORBフィールドを参照して、その内容がNULL、すなわちリンクされているORBがなければ、フラグNullflagをオンにして、ドアベルあるいはORBポインタへ書き込みがあるまで待つ。一方、リンクされているORBがあれば、ORBポインタで指されているORBのNext_ORBフィールドの値を、ORBポインタに書き込む（ステップS1910）。ORBリストにリンクされているORB全てについてプリフェッチキューに入れるまで、この手順を繰り返す。

【0086】このようにして、ORBリストのORBをプリフェッチキューにいれる。また、ステップS1906～S1908の処理は、図1においてORBディスパッチャとして説明した処理である。

【0087】（ライト実行エージェント）ライト実行エージェントは、まずライトプリフェッチキューに未処理のライトコマンドがあるか調べ（ステップS2001）、あればNextreadpointerの指すORBをライトプリフェッチキューから取り出す（ステップS2002）。

なおNextreadpointerはキューの先頭のORBを指し示すポインタである。ORBを読み出すと、そのORBのSeq_IDフィールドの値が、ライト実行エージェントが保持する変数Sequenceidentifierよりも小さい値であるか判定する（ステップS2003）。ただし、Seq_ID及びSequenceidentifierはともに有限の桁数である。そこで、Seq_ID及びSequenceidentifierがともにnビットで表わされる場合、ステップS2003における比較

では、 $((2 \times n - 1) < 0 (= 2 \times n))$ と定めておく。

【0088】通常のシーケンスであれば、ひとつのORBの処理終了後、Sequenceidentifierには後述のように1加算される。また、SequenceidentifierとORBのSeq_IDは同じ桁数であり、Seq_IDも1ずつ加算される。そのため、支障なく処理が進んでいる場合には、SequenceidentifierとORBのSeq_IDとは、ステップS2003において一致しているはずである。したがって、正常に処理が進めば、処理はステップS2003からステップS2004に移行する。

【0089】そこでは、クライアントがバッファを用意してあるか調べ(ステップS2004)、用意できたなら、ORBのDataDescriptorの値とライト実行エージェントが保持するNextexepointerに格納されたoffset値とを加えたアドレスから所定サイズのデータを読み、クライアントが用意したバッファに格納する(ステップS2005)。新たにORBの処理を行う場合には、offsetは0であるから、ORBにより指し示されるバッファの先頭から読み出すことになる。また、途中まで読み出されているバッファを続きから読む場合には、offsetは、バッファ中における続きのアドレスを示しているの、これを加算することで続きから読み出しを続行できる。データをバッファに格納したなら、DataDescriptorの値+offsetが次に読み出されるアドレスを指すように、Nextexepointerを更新する(ステップS2006)。このような読み出しを、ORBのdata_sizeで示されるサイズ分に達するまで行う(ステップS2007)。

【0090】こうして1つのORBの処理を終えると、Sequenceidentifierに1加算して更新し(ステップS2008)、Nextexepointer(offset)を0に初期化する(ステップS2009)。

【0091】処理を終えたなら、処理完了を通知するステータスブロックを作成し(ステップS2010)、ステータスFIFOに書き込み(ステップS2011)、ターゲットのクライアントに受信完了を通知する(ステップS2012)。

【0092】ステップS2003において、ORBのSeq_IDがSequenceidentifierよりも小さい場合は、バスリセットなどにより、イニシエータが図18の手順でORBリストを再生したような場合である。例えば、あるORBについて、ターゲットは処理を終え、Sequenceidentifierの値も更新していても、そのステータスブロックがイニシエータに届いていない時点でバスリセットが発生すると、イニシエータはそのORBもORBリストに入れてしまう。この場合に、ORBのSeq_IDよりも実行エージェントのSequenceidentifierの方が大きくなる。この場合には、そのORBは処理を終えているため、ステップS2010以降によりステータスブロックをイニ

シエータに渡す。

【0093】(リード実行エージェント)リード実行エージェントは、まずリードプリフェッチキューに未処理のリードコマンドがあるか調べ(ステップS2101)、あればNextreadpointerの指すORBをリードプリフェッチキューから取り出す(ステップS2102)。このNextreadpointerは、ライト実行エージェントのそれと同じ名称ではあるが、それとは独立して備えられている。なおNextreadpointerはキューの先頭のORBを指し示すポインタである。ORBを読み出すと、そのORBのSeq_IDフィールドの値が、リード実行エージェントが保持する変数Sequenceidentifierよりも小さい値であるか判定する(ステップS2103)。このSequenceidentifierも、ライト実行エージェントのそれと同じ名称ではあるが、それとは独立して有する。Seq_ID及びSequenceidentifierはともに有限の桁数である。そこで、Seq_ID及びSequenceidentifierがともにnビットで表わされる場合、ステップS2103における比較では、 $((2 \times n - 1) < 0 (= 2 \times n))$ と定めておく。

【0094】通常のシーケンスであれば、ひとつのORBの処理終了後、Sequenceidentifierには後述のように1加算される。また、SequenceidentifierとORBのSeq_IDは同じ桁数であり、Seq_IDも1ずつ加算される。そのため、支障なく処理が進んでいる場合には、SequenceidentifierとORBのSeq_IDとは、ステップS2003において一致しているはずである。したがって、正常に処理が進めば、処理はステップS2103からステップS2104に移行する。

【0095】そこでは、クライアントがデータを用意してあるか調べ(ステップS2104)、用意できたなら、ORBのDataDescriptorの値とリード実行エージェントが保持するNextexepointerに格納されたoffset値とを加えたアドレスに、クライアントのデータを所定サイズ書き込む(ステップS2105)。なお、このNextexepointerも、ライト実行エージェントのそれと同じ名称ではあるが、それとは別のポインタである。新たにORBの処理を行う場合には、offsetは0であるから、ORBにより指し示されるバッファの先頭から書き込むことになる。また、途中まで書き込まれているバッファを続きから書く場合には、offsetは、バッファ中における続きのアドレスを示しているの、これを加算することで続きから書き込みを続行できる。データをバッファに格納したなら、DataDescriptorの値+offsetが次に書き込まれるアドレスを指すように、Nextexepointerを更新する(ステップS2106)。このようなデータの書き込みを、クライアントが用意したデータを書き込み終わるまで行う(ステップS2107)。

【0096】こうして1つのORBの処理を終えると、

Sequenceidentifierに1加算して更新し(ステップS2108)、Nextexecpointer(offset)を0に初期化する(ステップS2109)。

【0097】処理を終えたなら、処理完了を通知するステータスブロックを作成し(ステップS2110)、ステータスFIFOに書き込み(ステップS2111)、ターゲットのクライアントに受信完了を通知する(ステップS2112)。

【0098】ステップS2103において、ORBのSeq_IDがSequenceidentifierよりも小さい場合は、バスリセットなどにより、イニシエータが図18の手順でORBリストを再生したような場合であるので、ライト実行エージェントと同様に、ステップS2110以降によりステータスブロックをイニシエータに渡す。

【0099】<ターゲットへのデータライトシーケンス>図22は、イニシエータ(ホストコンピュータ)からターゲット(プリンタ)へのライトを行う際のシーケンス例である。

【0100】なお、図中のSBP-2とは、SBP-2規格で規定されたデータ、すなわちSBP-2で規定されたフィールドを処理する処理層である。また、SHPTとは、SBP-2で規定されない、ファンクションごとに定められた処理を行う処理層である。SHPTでは、前述したフローチャートの手順を行う。SBP-2は、その手順において、ORBのリンクやドアベルのライト、ORBやステータスのSHPTへの引き渡し等の機能を果たす。

【0101】まず、イニシエータのクライアントでデータ及びライトコマンドが、SHPTでORB(ここではライトORB)が作成され、カウンタCurrentWriteQUEが1減らされて、ORBのリンク要求がSBP-2に出される(1901)。SBP-2ではORBをリンクしてドアベルレジスタにライト要求を出す(1902)。1394インタフェースにより、ドアベルレジスタが書き込まれ(1903)、それがターゲットのSBP-2に通知される(1904)。

【0102】通知を受けたSBP-2は、ORBリード要求を1394インタフェースに出して(1905)、システムメモリからORBが読み出される(1906)。SHPTは、読み出されたORBの内容に応じて対応キューに格納する(1907)。ここではライトコマンドORBなので、1394インタフェースに対してデータリード要求を出す(1908)。これに応じて、指定されたアドレスからデータが読み出され、クライアントに渡される(1909)。クライアントは、この場合画像の展開を行うラスタライザ等であり、データが展開された画像であればそれはプリンタエンジンから印刷出力されることになる。

【0103】処理が済むとライト実行エージェントからSBP-2に対してステータスブロック送信要求が出さ

れ(1910)、ステータスブロックがイニシエータに返される(1911)。イニシエータのSHTPは、これを受けると対応ORBをリンクから削除し、空きキューカウンタを1増やす(1912)。

【0104】以上のシーケンスによりイニシエータからターゲットにデータが書き込まれる。

【0105】<ターゲットからのデータリードシーケンス>図23は、イニシエータ(ホストコンピュータ)からターゲット(プリンタ)のデータのリードを行う際のシーケンス例である。

【0106】まず、イニシエータのクライアントでリードコマンドが、SHPTでORB(ここではリードORB)が作成され、カウンタCurrentReadQUEが1減らされて、ORBのリンク要求がSBP-2に出される(1901)。SBP-2ではORBをリンクしてドアベルレジスタにライト要求を出す(1902)。1394インタフェースにより、ドアベルレジスタが書き込まれ(1903)、それがターゲットのSBP-2に通知される(1904)。

【0107】通知を受けたSBP-2は、ORBリード要求を1394インタフェースに出して(1905)、システムメモリからORBが読み出される(1906)。SHPTは、読み出されたORBの内容に応じて対応キューに格納する(1907)。

【0108】リード実行エージェントは、ここでイニシエータに渡すデータが発生するまで待つ。データが発生すると、1394インタフェースに対してデータライト要求を出す(2008)。これに応じて、指定されたアドレスに用意されたデータが書き込まれる(2009)。

【0109】処理が済むとリード実行エージェントからSBP-2に対してステータスブロック送信要求が出され(1910)、ステータスブロックがイニシエータに返される(1911)。イニシエータのSHTPは、これを受けると対応ORBをリンクから削除し、空きキューカウンタを1増やす(1912)。

【0110】このシーケンスによりイニシエータからターゲットのデータが読み出される。

【0111】以上のようにして、イニシエータとターゲットとの間で、簡素な制御手順によりデータの交換が双方向に行える。すなわち、イニシエータは所望のときに所望のデータをターゲットに渡すことができる。また、ターゲットは、イニシエータから渡されたデータをターゲット自身の都合に合わせて読み出すことができる。また、ターゲットはイニシエータが準備していさえすれば、自発的であろうと、あるいはイニシエータからの要求であろうと、何時でもイニシエータにデータを渡すことができる。また、バスリセットが発生しても、バスリセット直前の状態からの処理の続行を保証することができる。

10

20

30

40

50

【0112】

【他の実施形態】さらに、図7に示したように、ORBにはPSID、SSIDというチャンネル識別子のフィールドを備えている。そのため、SHP T処理部及びORBディスパッチャがチャンネル識別子を識別して、チャンネルごとに独立したキュー及び実行エージェントを用いてORBの生成・処理を行えば、マルチチャンネルを実現することができる。この場合、1つの機器であっても、そこに含まれる複数のクライアントそれぞれに1つのチャンネルを割り当てることで、クライアントごとに非同期の通信を行わせることができる。このため、例えばデジタル複合機であれば、それが有するスキャナやプリンタそれぞれについてクライアントとなるアプリケーションを設けておけば、それと接続されたホストコンピュータからは、あたかもそれぞれの機能を独立した機器であるかのように使用することができる。フローチャートで示したように、各キューの管理や実行エージェントは独立したプロセスであるために、マルチチャンネル化は容易に行うことができる。

【0113】なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェイス機器、リーダ、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0114】また、本発明の目的は、イニシエータとして図13～図18の手順のプログラムコードを記録した記憶媒体をコンピュータに供給し、そのコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても達成される。またターゲットも、図19～図21の手順のプログラムコードを記録した記憶媒体をコンピュータに供給し、そのコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによって実現できる。

【0115】この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0116】プログラムコードを供給するための記憶媒体としては、例えば、フロッピディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

【0117】また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【0118】さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【0119】また、記憶媒体から読み出されたプログラムコードを実行してイニシエータとなる主体はコンピュータに限らず、データ転送機器であって、SBP-2のログイン能力があれば、どのような機器であっても良い。

【0120】

【発明の効果】以上説明したように、本発明によれば、イニシエータとターゲットとの間において、1回のログインで非同期な双方向通信を可能とし、また、データの交換に必要なプロセスやメモリといった資源を効率的に利用できる。

【0121】また、IEEE1394インターフェースを利用しているために、ターゲット側へのデータ転送を、ターゲットがその都合に応じて読み出すことで行うことができ、イニシエータがターゲットの都合によってデータ転送に占有されることを防止できる。

【0122】また、SBP-2プロトコルを利用しているため、ターゲットでキューされるのはORBだけで、実際に転送されるデータそのものは処理待ちの間イニシエータに格納されている。このため、ターゲットのメモリ資源を小容量化できる。

【0123】また、簡単にマルチチャンネルを実現できる。

【0124】また、最新の処理状態を保持することで、バスリセットが発生しても、バスリセット後に、バスリセット直前の状態から処理を再開することができ、処理の正常な続行を保証することができる。

【0125】

【図面の簡単な説明】

【図1】ターゲット（プリンタ）のブロック図である。

【図2】イニシエータ（ホストコンピュータ）のブロック図である。

【図3】イニシエータとターゲットとの間におけるキューの状態の一例を示す図である。

【図4】イニシエータにおけるバスリセット後の復旧の様子の例を示す図である。

【図5】ターゲットにおけるバスリセット後の復旧の様子の例を示す図である。

【図6】イニシエータにおいてORBリストからORBを取り外す様子の例を示す図である。

【図7】ORB一般のフォーマットを示す図である。

【図8】ライトコマンドORBのフォーマットを示す図

である。

【図9】リードコマンドORBのフォーマットを示す図である。

【図10】ステータスブロック一般のフォーマットを示す図である。

【図11】ライトステータスブロックのフォーマットを示す図である。

【図12】リードステータスブロックのフォーマットを示す図である。

【図13】イニシエータのクライアントによる、データ転送要求の発生時の処理手順のフローチャートである。

【図14】イニシエータのクライアントによる、SHTPからの処理終了通知受信時の処理手順のフローチャートである。

【図15】イニシエータのSHTP処理部により実行される、クライアントからコールされるライト関数及びリード関数のフローチャートである。

【図16】イニシエータのSHTP処理部により実行される、ライトI/O要求キュー及びリードI/O要求キューの管理手順のフローチャートである。

【図17】イニシエータのSHTP処理部により実行される、ステータスブロック受信時の処理手順のフローチャートである。

【図18】イニシエータのSHTP処理部により実行される、バスリセット直後の復旧処理手順のフローチャートである。

【図19】ドアベルレジスタあるいはORBポインタに書き込みが行われた際のターゲットのフェッチエージェントによる処理手順のフローチャートである。

【図20】ライト実行エージェントによる処理手順のフローチャートである。

【図21】リード実行エージェントによる処理手順のフ

* ローチャートである。

【図22】イニシエータ（ホストコンピュータ）からターゲット（プリンタ）へのデータライト時のシーケンスの図である。

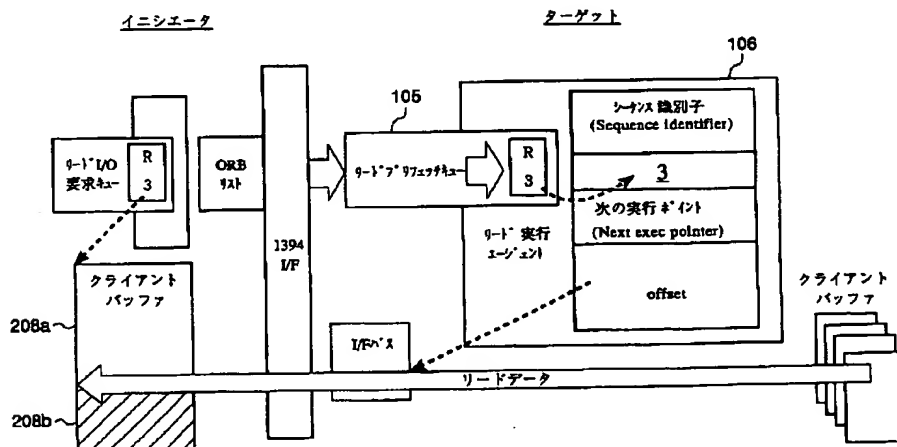
【図23】イニシエータ（ホストコンピュータ）からターゲット（プリンタ）へのデータリード時のシーケンスの図である。

【図24】IEEE1394インターフェースを用いたプリンタシステムのハードウェア構成図である。

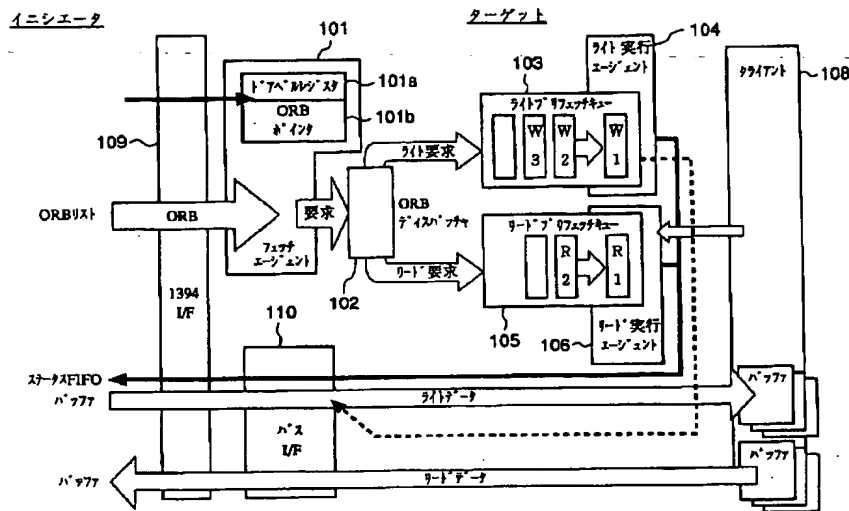
【符号の説明】

- 1 CPU
- 2 RAM
- 3 ROM
- 4 システムバス
- 12 CPU
- 13 ROM
- 19 RAM
- 101 フェッチエージェント
- 103, 105 プリフェッチキュー
- 104 ライト実行エージェント
- 106 リード実行エージェント
- 108 ターゲット側クライアント
- 109 1394インターフェース
- 110 バスインターフェース
- 201 SHTP処理部
- 202 ライトI/O要求キュー
- 203 リードI/O要求キュー
- 204 ORBリスト
- 205 ステータスFIFO
- 206 イニシエータ側クライアント
- 207 ライトバッファ
- 208 リードバッファ

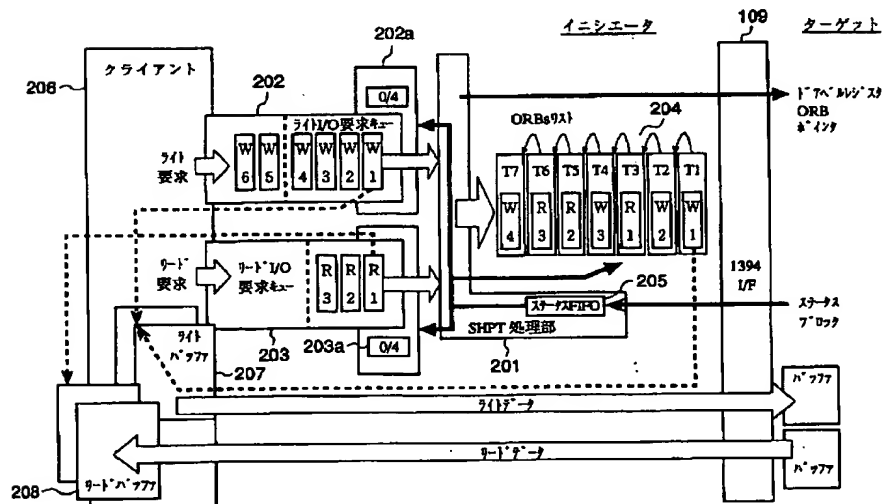
【図5】



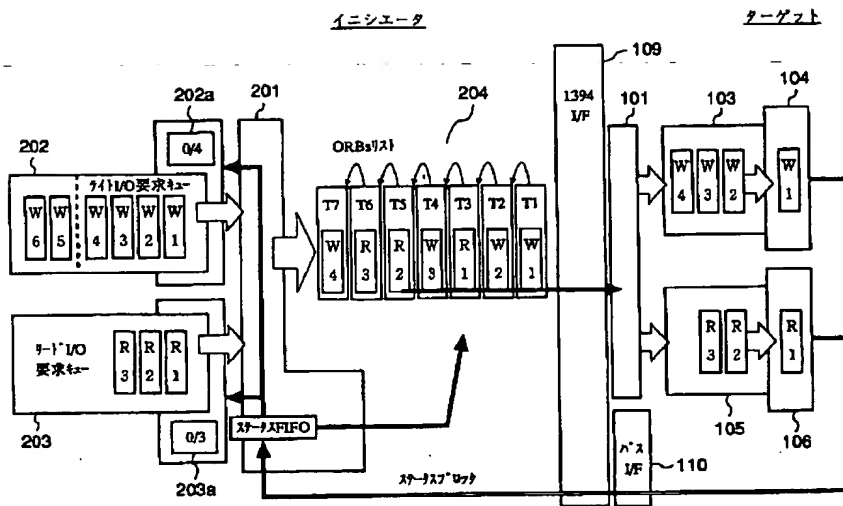
【図1】



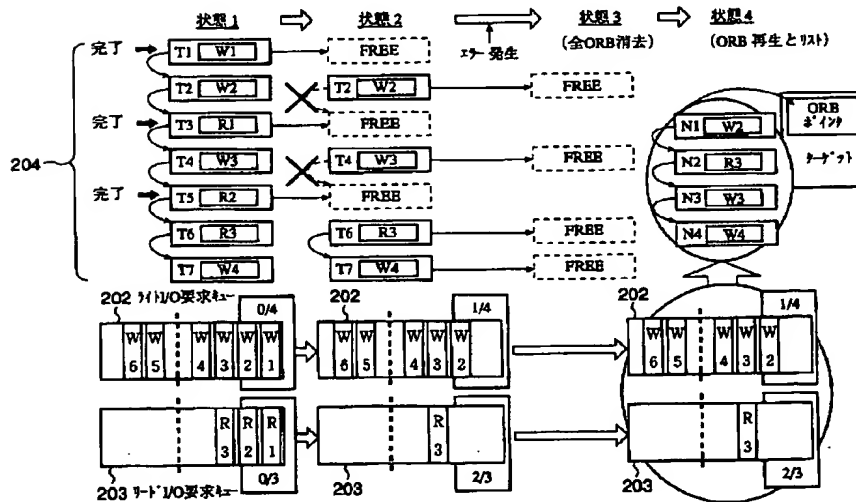
【図2】



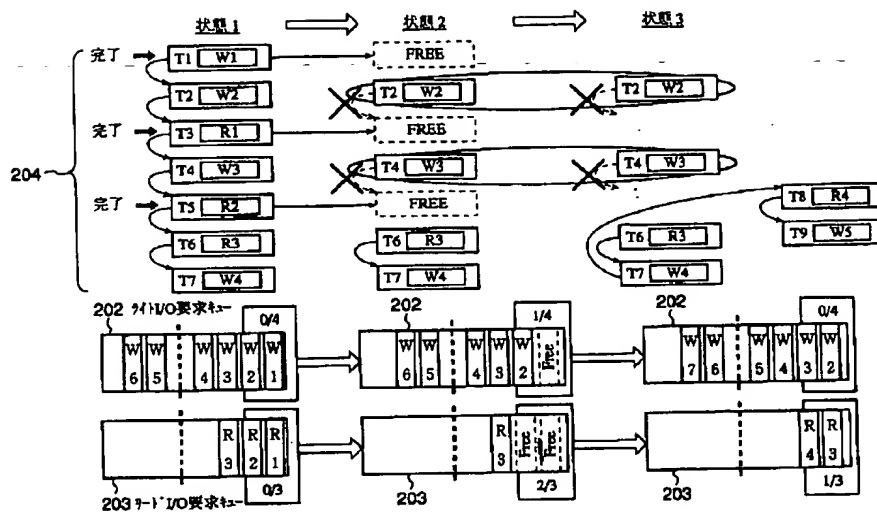
【図3】



【図4】

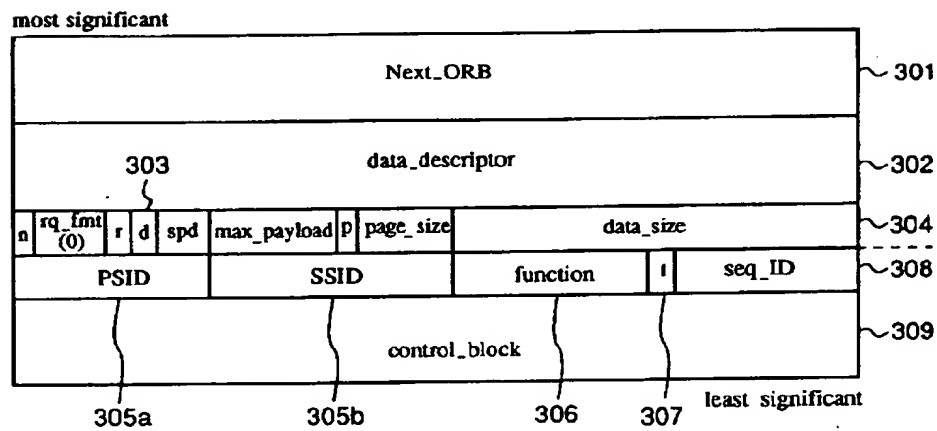


【図6】



【図7】

(A)



(B)

Value	SHPT コマンド
0	ライト
1-3F ₁₆	reserved for future standardization
40 ₁₆	リード
41 ₁₆ -FF ₁₆	reserved for future standardization

【図8】

most significant

Next_ORB										
data_descriptor										
n	rq_fmt (0)	r	d	spd	max_payload	P	page_size	data_size		
PSID				SSID			Function (0)		t	seq_ID
Reserved										

least significant

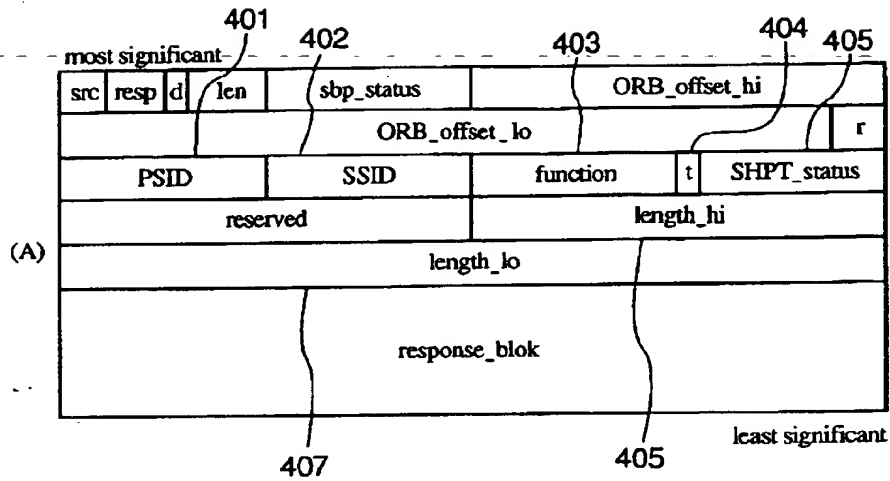
【図9】

most significant

Next_ORB											
data_descriptor											
n	rq_fmt (0)	r	d	spd	max_payload	p	page_size	data_size			
PSID				SSID			Function (4016)		t	seq_ID	
Reserved											

least significant

【図10】



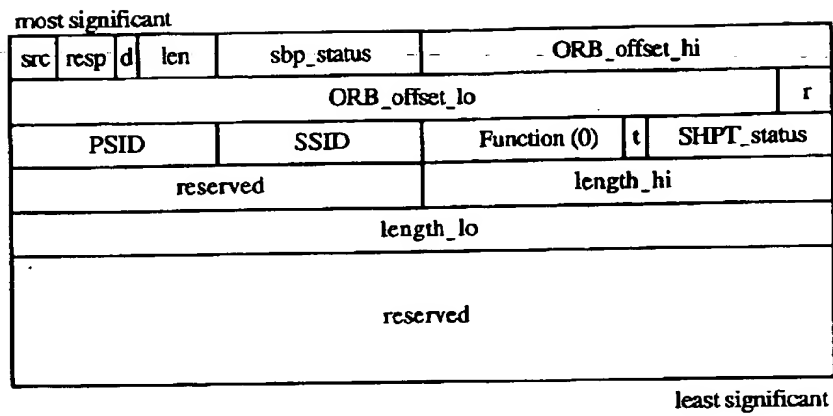
(B)

Value	SHPT ファンクション
0	ライト
1 - 3F ₁₆	reserved for future standardization
40 ₁₆	リード
41 ₁₆	Data available
42 ₁₆ - FF ₁₆	reserved for future standardization

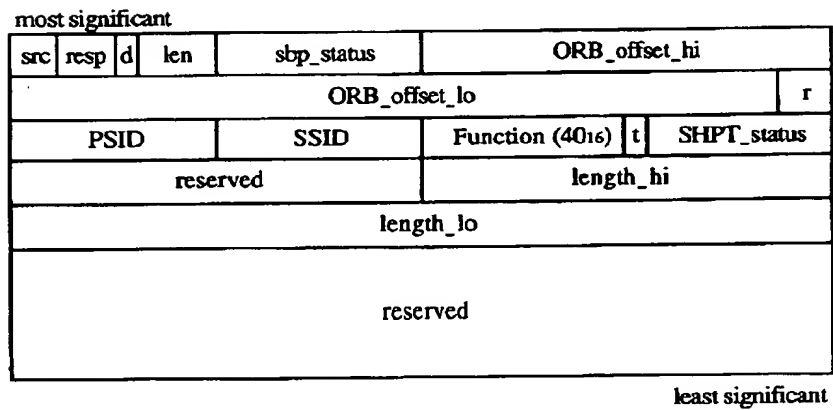
(C)

Value	SHPT_status
0	完了
1	部分終了
2	エラー
3 - 7F ₁₆	reserved for future standardization

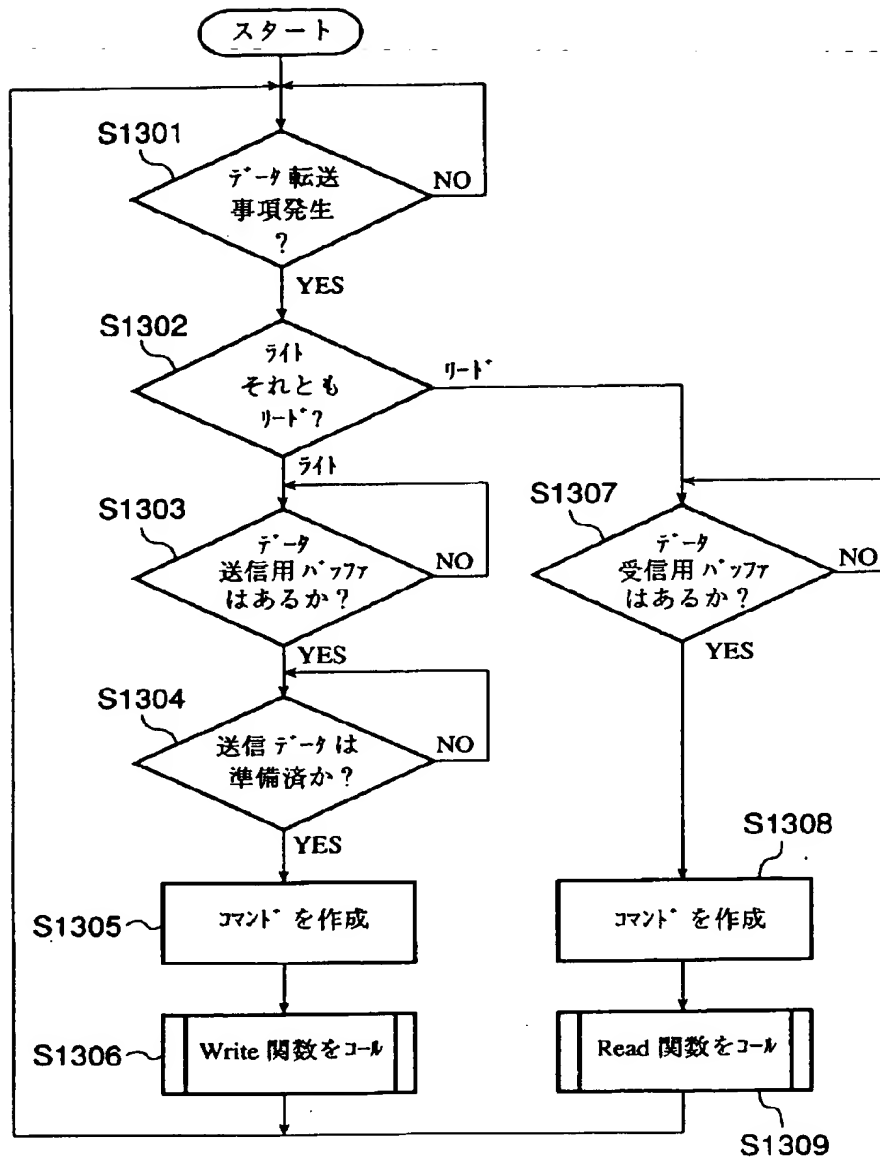
【図 1 1】



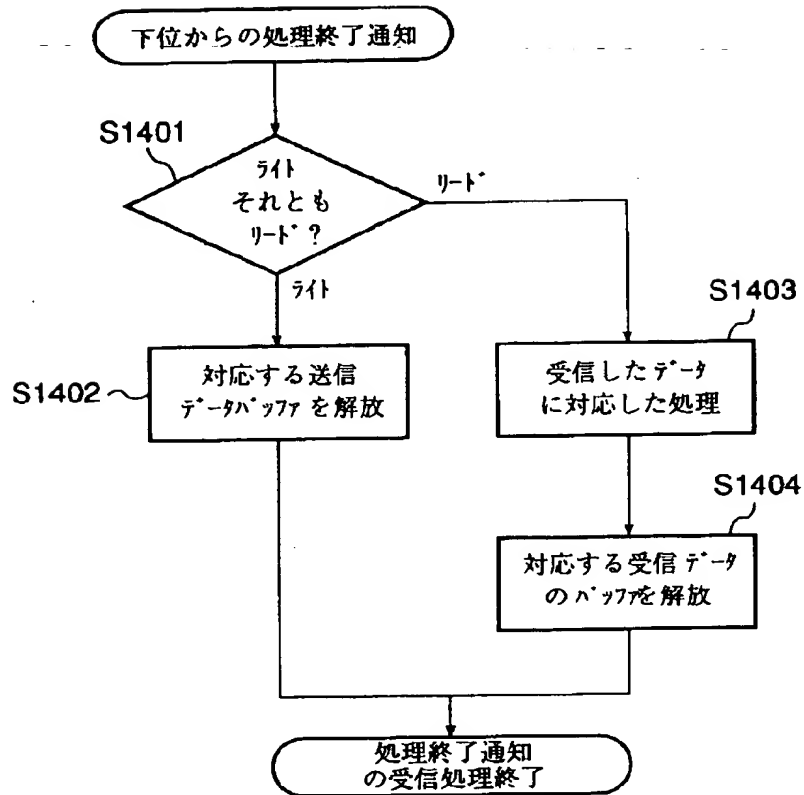
【図 1 2】



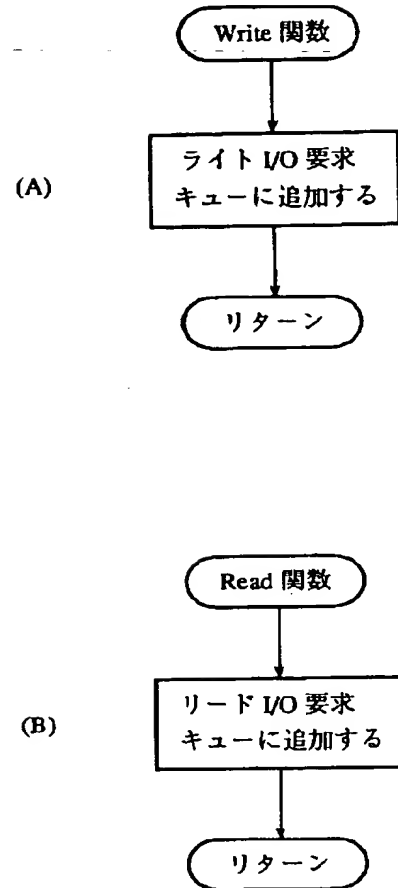
【図13】



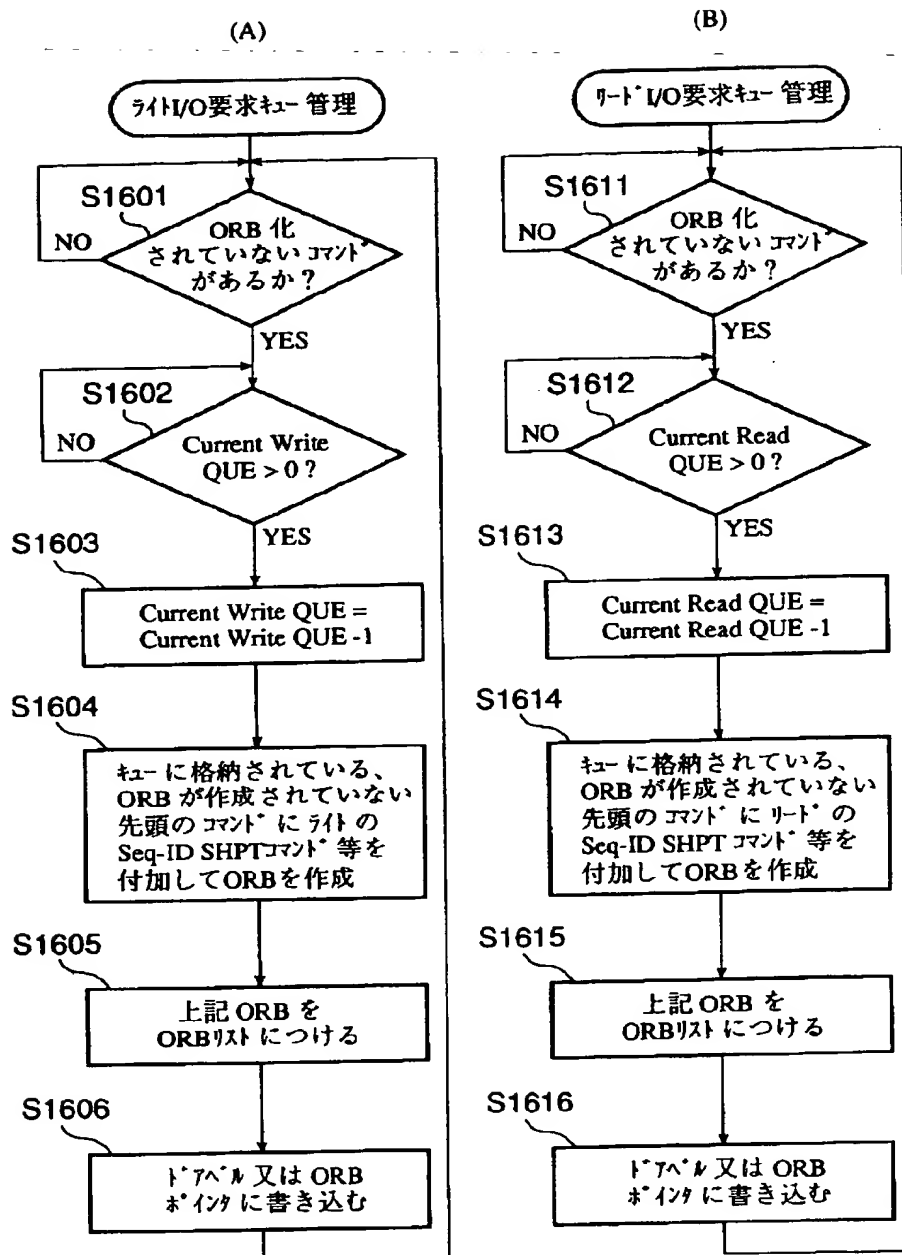
【図14】



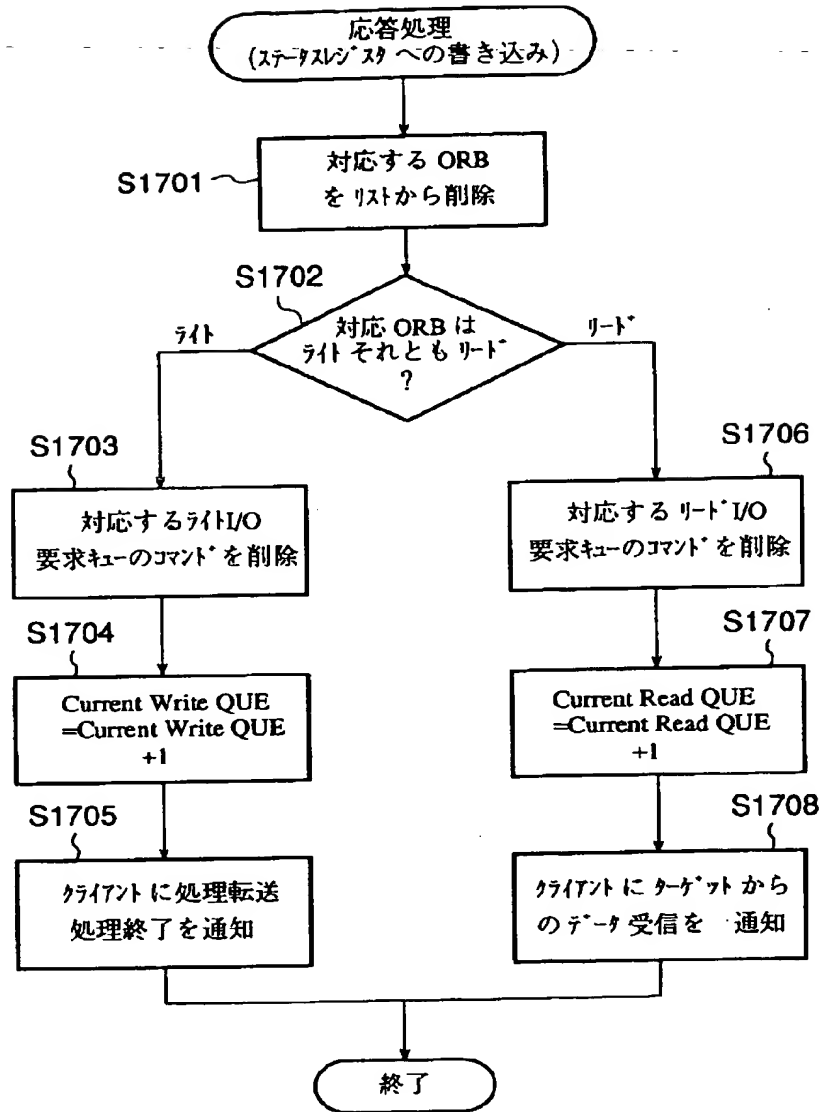
【図15】



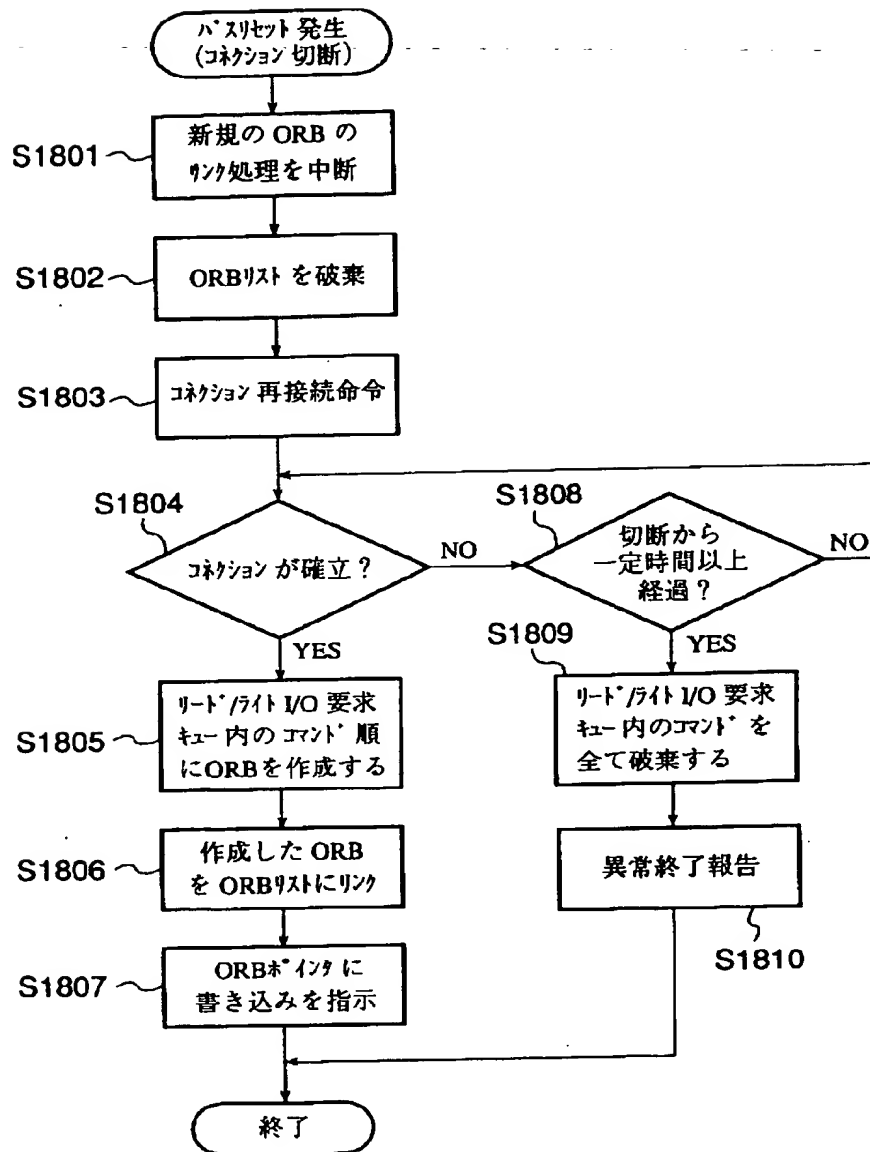
【図16】



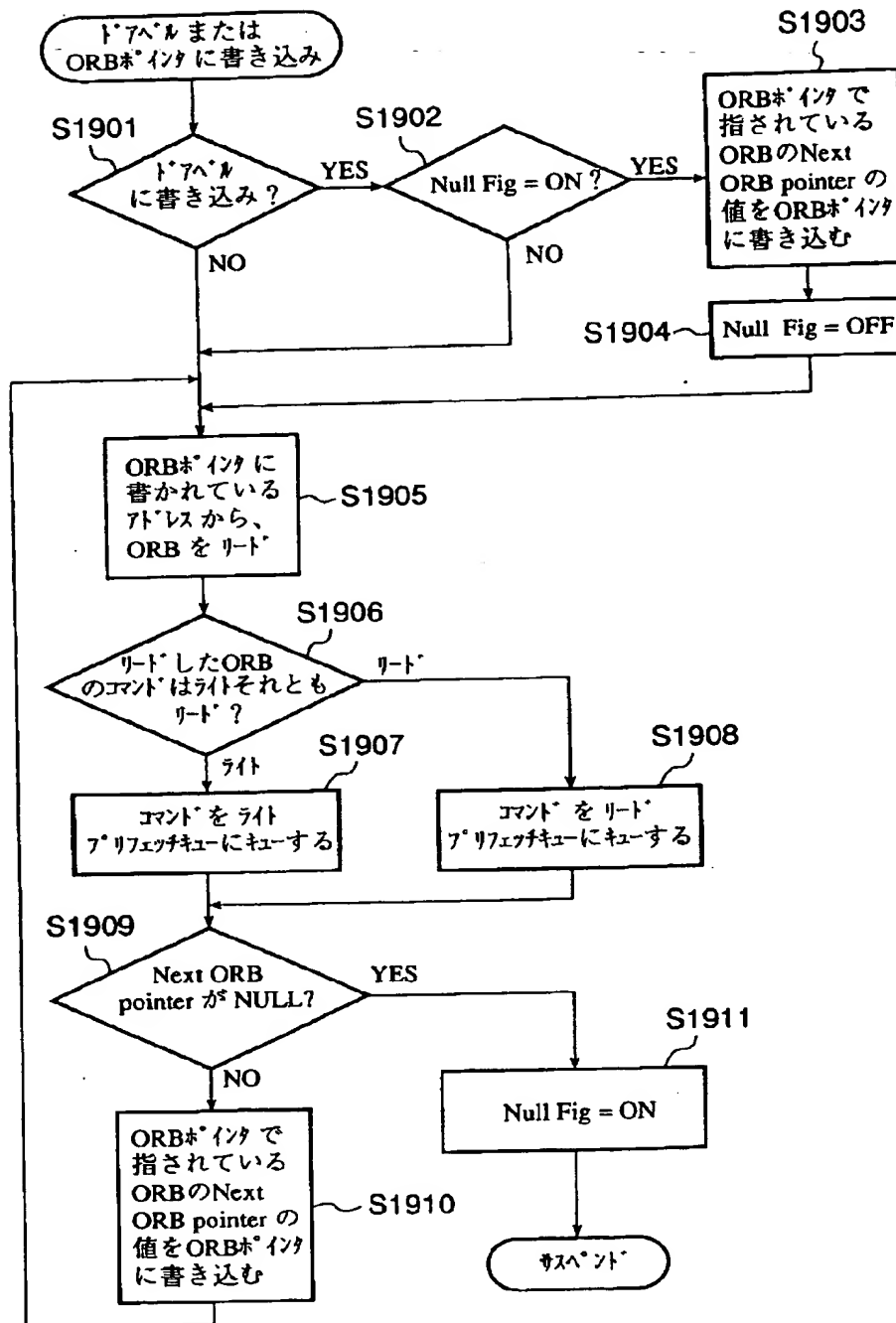
【図17】



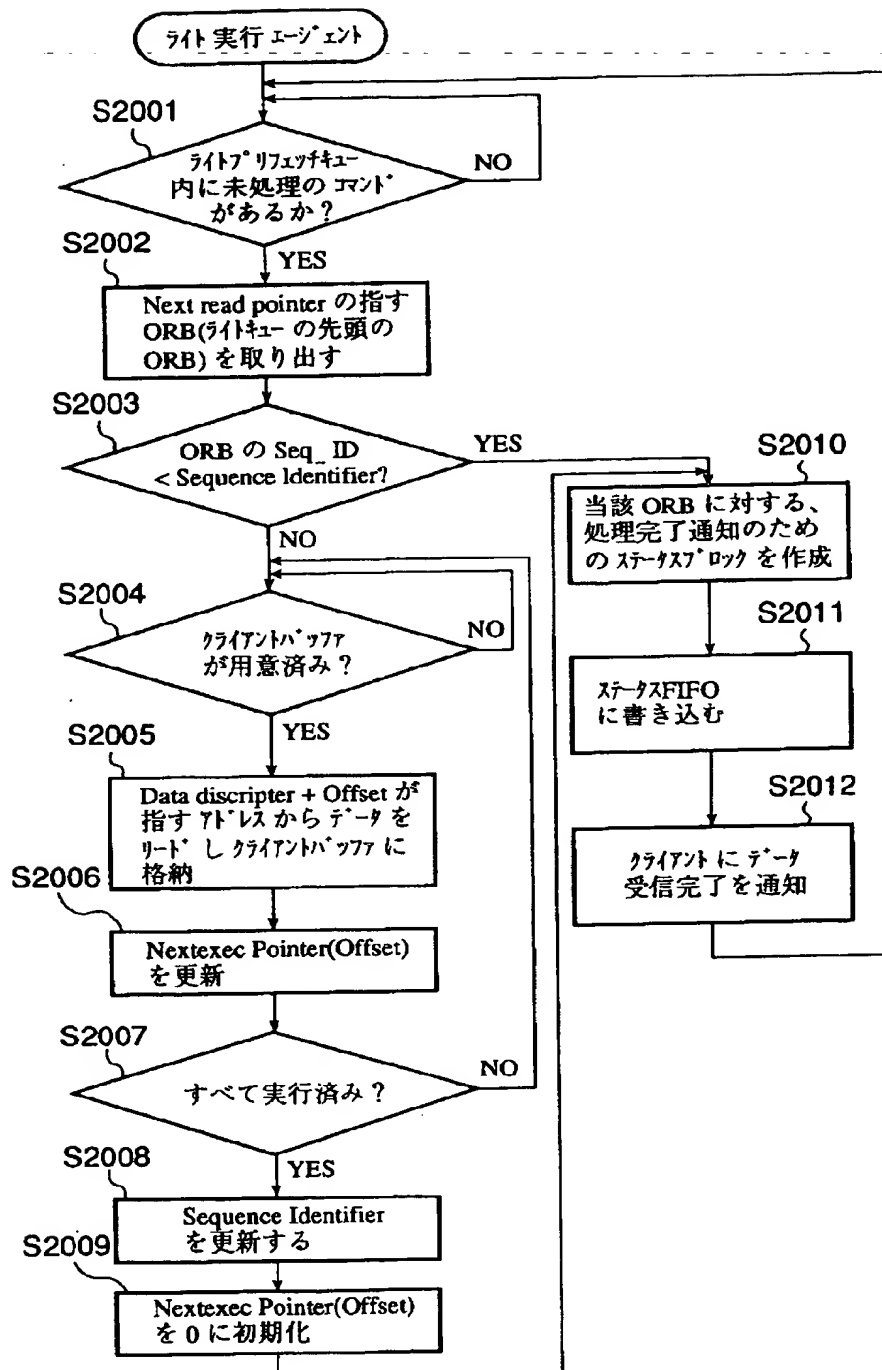
【図18】



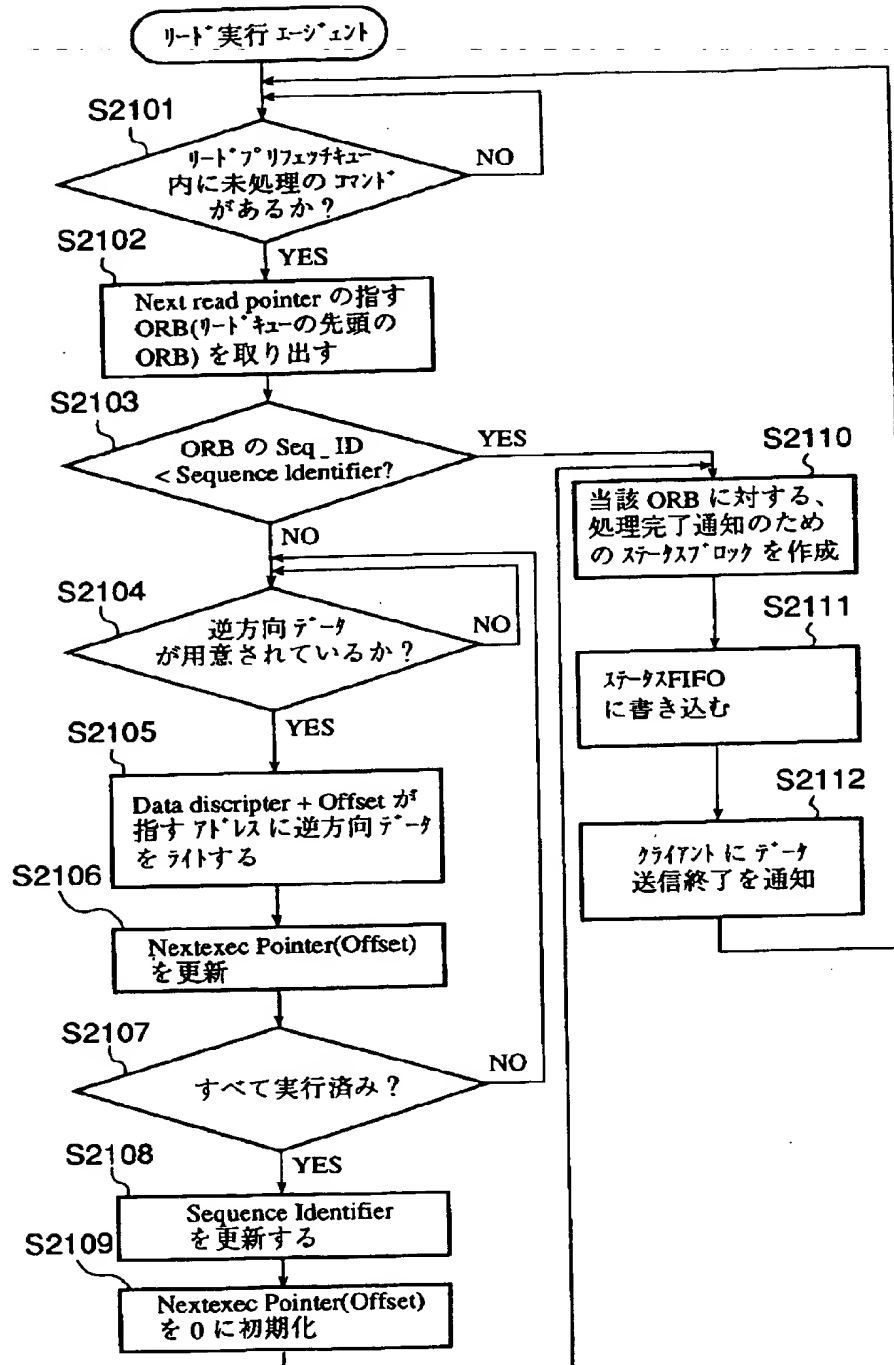
【図19】



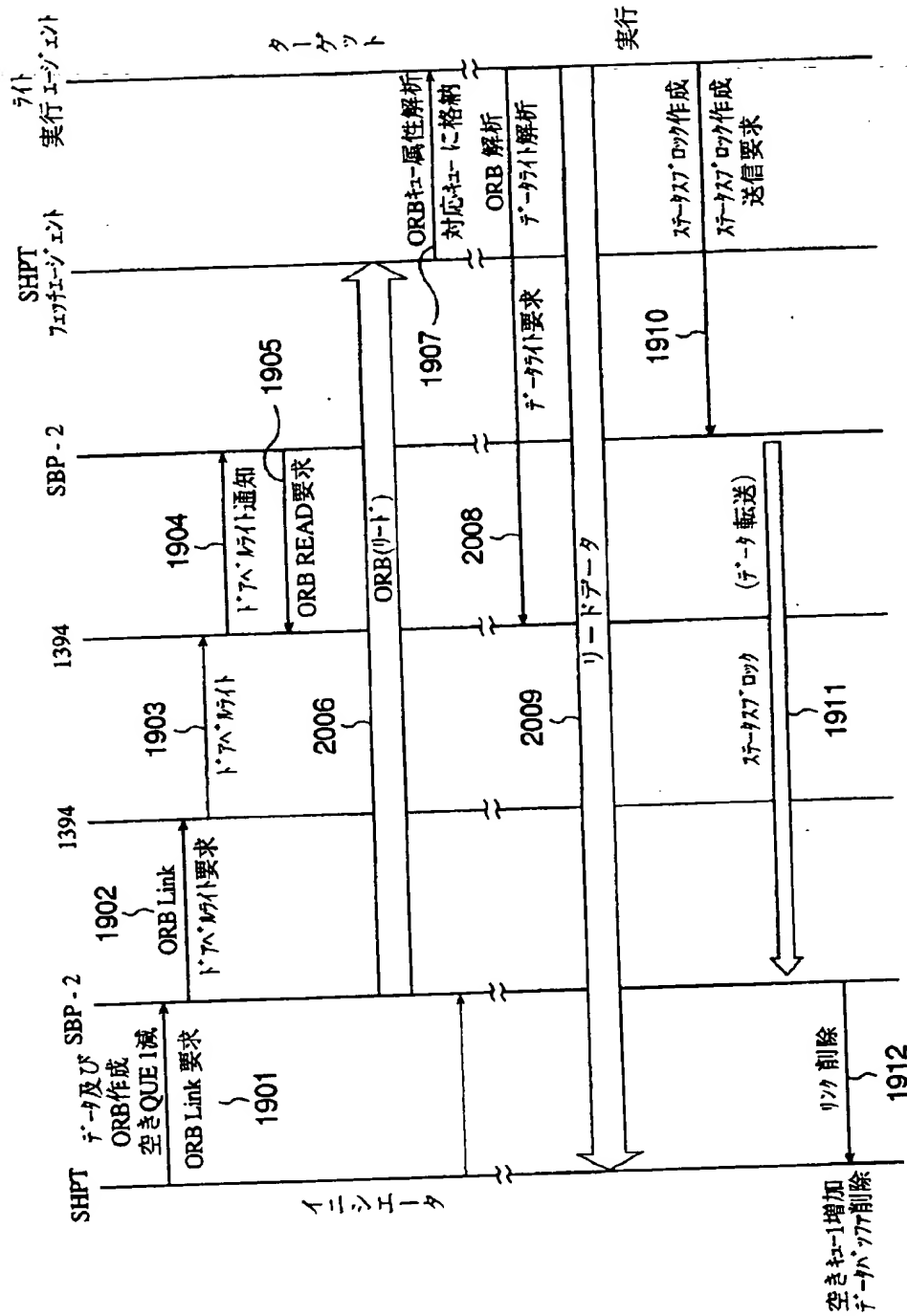
【図20】



【図21】



【図23】



【図24】

